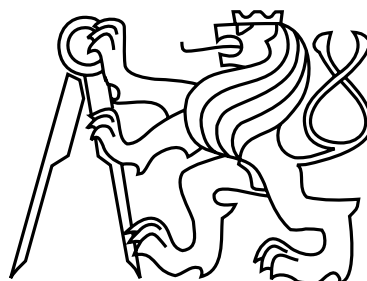


České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Diplomová práce

Parametrizovaný generátor náhodných logických obvodů

Bc. Karel Houžvička

Vedoucí práce: Ing. Petr Fišer Ph.D.

Studijní program: Elektrotechnika a informatika, dobíhající, Magisterský

Obor: Výpočetní technika

6. května 2012

Poděkování

Velice děkuji své rodině a přátelům za podporu a povzbuzení a v situacích, kdy se cesta k cíli zdála až příliš klikatá. Velké poděkování patří i vedoucímu práce panu Ing. Petru Fišerovi Ph.D., za vzorné vedení a rady.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 7. 5. 2012

.....

Abstract

The goal of the thesis is to program a parameterised generator of logic circuits which will be able to save the results to BLIF format. An intention of this work will be also to detect dependencies of parameters, which can influence a circuit minimisation by synthesis tools. After that analyse standard test circuits and due to selected parameters create clones by developed program. Finally compare the origin circuits with this clones.

Abstrakt

Cílem této práce je naprogramovat parametrizovaný generátor logických obvodů a ukládat výsledné obvody do formátu BLIF. Účelem práce je také zjistit závislosti jednotlivých parametrů na celkovou minimalizovatelnost obvodu pomocí syntézních nástrojů. Poté analyzovat standardní testovací obvody a pomocí získaných parametrů vytvořit výsledným programem klony. Ty následně experimentálně porovnat s originály.

Obsah

1	Úvod	1
2	Popis problému a specifikace cíle	3
2.1	Logická hradla	3
2.2	Počet vstupních portů	4
2.3	Počet výstupních portů	4
2.4	Počet úrovní obvodu	4
2.5	Maximální návaznost signálů	4
2.6	Celkový počet hradel obvodu	5
2.7	Maximální počet vstupů hradla	5
2.8	Maximální a průměrné větvení signálu	5
2.9	Počet úrovní obvodu mezi vstupy a výstupem	6
2.10	Rekonvergence	6
2.11	Sekvenční obvody	7
2.12	Ostatní podmínky	7
3	Analýza problémů a stanovení postupu	9
3.1	Základní pohled na problematiku	9
3.2	Generování hradla	10
3.2.1	Výpočet říditelnosti	10
3.3	Generování mřížky hradel	11
3.4	Přesný počet hradel	12
3.4.1	Závislost počtu hradel na šířce mřížky	16
3.4.2	Závislost počtu hradel na počtu vstupních signálů	16
3.4.3	Závislost na maximální návaznosti signálů	17
3.4.4	Závislost na maximálním počtu vstupů hradla	18
3.4.5	Heuristika nastavení šířky mřížky	19
3.5	Vytvoření více výstupů	22
3.6	Pojmenování signálů	24
3.7	Snížení maximálního a průměrného větvení signálu	24
3.7.1	Maximální větvení signálu	24
3.7.2	Průměrné větvení signálu	25
3.8	Hledání rekonvergenčí	26
3.9	Sekvenční obvody	28

4	Implementace	29
4.1	Programovací jazyk	29
4.2	Výstupní formát	29
4.3	Vstupní formát	30
4.4	Hlavní datové struktury	30
4.5	Popis tříd	31
4.5.1	Class Hradlo	31
4.5.2	Class Gener	31
4.5.3	Class GenerVstupu	32
4.5.4	Class Vstup	32
4.5.5	Class RekoObj	32
4.5.6	Class RekonBunkaSeznam	32
4.5.7	Class RekBunka	33
4.6	Náhodný generátor vstupů hradla	33
4.7	Generování mřížky	35
4.8	Hledání přesného počtu hradel	36
4.9	Dovytyvoření výstupů	37
4.10	Úprava větvení signálů	38
4.11	Výpočet smyček rekonvergenčí	40
4.12	Získávání informací	42
4.13	Porovnání s existujícím řešením	44
5	Testování	45
5.1	Testování rekonvergence	45
5.1.1	Závislost počtu smyček rekonvergenčí na parametrech obvodu	45
5.1.2	Projev snížení maximálního větvení signálu na počtu smyček rekonvergenčí	48
5.1.3	Zhodnocení	50
5.2	Test vstupních parametrů v závislosti na minimalizovatelnosti	51
5.2.1	Metodika	51
5.2.2	Závislost na počtu vstupů	51
5.2.3	Závislost na počtu výstupů	52
5.2.4	Závislost na maximální návaznosti signálů	52
5.2.5	Závislost na maximálním počtu vstupů hradla	53
5.2.6	Závislost na použití hradel XOR a XNOR	53
5.2.7	Závislost na počtu vstupních signálů při nejhorší možnosti	54
5.2.8	Zhodnocení	54
5.3	Klony standardních zkušebních obvodů	55
5.3.1	Metodika	55
5.3.2	Obvod apex6.blif	56
5.3.3	Obvod apex7.blif	58
5.3.4	Obvod b9.blif	60
5.3.5	Obvod c880.blif	62
5.3.6	Obvod c1908.blif	64
5.3.7	Obvod count.blif	66
5.3.8	Obvod term1.blif	68
5.3.9	Obvod too-large.blif	70

5.3.10	Obvod dk15.blif	72
5.3.11	Obvod tav.blif	74
5.3.12	Obvod train11.blif	76
5.3.13	Zhodnocení	77
5.4	Shrnutí dodržování požadovaných parametrů obvodu	78
6	Závěr	81
A	Naměřená data	85
B	Formát BLIF	91
C	Instalační a uživatelská příručka	93
C.1	Instalace	93
C.2	Uživatelská příručka	93
C.2.1	Formát vstupního souboru	94
D	Obsah přiloženého CD	97

Seznam obrázků

2.1	Ukázka úrovní obvodu	4
2.2	Ukázka návaznosti signálů	5
2.3	Ukázka rekonvergence	6
2.4	Ukázka obvodu se smyčkami rekonvergence	7
3.1	Struktura generování obvodu	11
3.2	Četnost počtu hradel na úrovni	13
3.3	Četnost počtu hradel pro dva vstupy	15
3.4	Závislost počtu hradel na šířce mřížky	16
3.5	Závislost počtu hradel na počtu vstupů obvodu	17
3.6	Závislost počtu hradel na návaznosti signálů	18
3.7	Závislost počtu hradel na max počtu vstupů hradla	19
3.8	Počáteční bod lineární závislosti	20
3.9	Zobrazení rozptylu hodnot okolo průměrné hodnoty	21
3.10	Vytvoření více výstupů	23
3.11	Snížení větvení	24
3.12	Ukázka výpočtu rekonvergence	27
3.13	Princip sekvenčního obvodu	28
5.1	Graf rozložení velikosti rekonvergenčí pro nastavení M1 až M4	47
5.2	Graf rozložení velikosti rekonvergenčí pro nastavení M5 až M8	48
5.3	Změna smyček rekonvergence při návaznosti 1 a 20	50

Seznam tabulek

2.1	Tabulka typů hradel	3
3.1	Tabulka parametrů	13
3.2	Tabulka naměřených průměrů	14
3.3	Tabulka naměřených průměrů, parametry měření	15
3.4	Naměřené změny šířky mřížky v závislosti na maximálním počtu vstupu hradla	20
5.1	Parametry obvodů rekonvergence	46
5.2	Počet rekonverzí	46
5.3	Průměrné snížení maximálního větvení signálů	49
5.4	Závislost počtu rekonverzí na snížení větvení	49
5.5	Průměrné hodnoty minimalizovaných obvodů, změna počtu vstupů	52
5.6	Průměrné hodnoty minimalizovaných obvodů, změna počtu výstupů	52
5.7	Průměrné hodnoty minimalizovaných obvodů, změna maximální návaznosti signálů	53
5.8	Průměrné hodnoty minimalizovaných obvodů, změna maximálního počtu vstupů hradla	53
5.9	Průměrné hodnoty minimalizovaných obvodů, použití hradel XOR a NXOR	54
5.10	Průměrné hodnoty minimalizovaných obvodů, nejhorší případ	54
5.11	Stejně parametry obvodu apex6.blif s klony	56
5.12	Parametry obvodu apex6.blif a jeho klonů	56
5.13	Parametry po minimalizaci souboru apex6.blif a jeho klonů	57
5.14	Stejně parametry obvodu apex7.blif s klony	58
5.15	Parametry obvodu apex7.blif a jeho klonů	58
5.16	Parametry po minimalizaci souboru apex7.blif a jeho klonů	59
5.17	Stejně parametry obvodu b9.blif s klony	60
5.18	Parametry obvodu b9.blif a jeho klonů	60
5.19	Parametry po minimalizaci souboru b9.blif a jeho klonů	61
5.20	Stejně parametry obvodu c880.blif s klony	62
5.21	Parametry obvodu c880.blif a jeho klonů	62
5.22	Parametry po minimalizaci souboru c880.blif a jeho klonů	63
5.23	Stejně parametry obvodu c1908.blif s klony	64
5.24	Parametry obvodu c1908.blif a jeho klonů	64
5.25	Parametry po minimalizaci souboru c1908.blif a jeho klonů	65
5.26	Stejně parametry obvodu count.blif s klony	66
5.27	Parametry obvodu count.blif a jeho klonů	66
5.28	Parametry po minimalizaci souboru count.blif a jeho klonů	67

5.29	Stejné parametry obvodu term1.blif s klony	68
5.30	Parametry obvodu term1.blif a jeho klonů	68
5.31	Parametry po minimalizaci souboru term1.blif a jeho klonů	69
5.32	Stejné parametry obvodu too-large.blif s klony	70
5.33	Parametry obvodu too-large.blif a jeho klonů	70
5.34	Parametry po minimalizaci souboru too-large.blif a jeho klonů	71
5.35	Stejné parametry obvodu dk15.blif s klony	72
5.36	Parametry obvodu dk15.blif a jeho klonů	72
5.37	Parametry po minimalizaci souboru dk15.blif a jeho klonů	73
5.38	Stejné parametry obvodu tav.blif s klony	74
5.39	Parametry obvodu tav.blif a jeho klonů	74
5.40	Parametry po minimalizaci souboru tav.blif a jeho klonů	75
5.41	Stejné parametry obvodu train11.blif s klony	76
5.42	Parametry obvodu train11.blif a jeho klonů	76
5.43	Parametry po minimalizaci souboru train11.blif a jeho klonů	77
5.44	Průměrná procentuální odchylka parametrů	79
5.45	Průměrná absolutní odchylka parametrů	80
A.1	Závislost průměrného počtu hradel na šířce mřížky graf 3.4	85
A.2	Parametry k tab A.1	85
A.3	Data, závislost na počtu vstupních signálů graf 3.5	86
A.4	Parametry k tab A.3	86
A.5	Data, závislost na návaznosti vstupních signálů graf 3.6	86
A.6	Parametry k tab A.5	86
A.7	Data, závislost na maximálním počtu vstupů hradla graf 3.7	87
A.8	Parametry k tab A.7	87
A.9	Data, počáteční bod charakteristiky graf 3.8	87
A.10	Parametry k tab A.9	87
A.11	Rozptyl kolem průměru hodnot	87
A.12	Data testování závislosti počtu vstupů při PO=5, N=10, vst. hradla=2	88
A.13	Data testování závislosti počtu výstupů při PI=10, N=10, vst. hradla=2	88
A.14	Data testování závislosti na návaznosti signálů při PI=10, PO=5, vst. hradla=2	89
A.15	Data testování závislosti max. počtu výstupů hradla při PI=10, PO=5, N=10	89
A.16	Data testování výskytu hradel XOR při PI=10, PO=5, N=10, vst. hradla=2, vst. hradla=2	90
A.17	Data testování nejhorší možnosti nastavení při PO=10, N=10, vst. hradla=4, s XOR	90

Kapitola 1

Úvod

Cílem diplomové práce je vytvořit generátor logických obvodů, který bude generovat obvody náhodně a vždy s co nejpřesnějším dodržením zadaných parametrů. Bude se jednat o víceúrovňovou síť hradel vytvořenou ze základních logických prvků. Výstupní soubor generátoru bude ve formátu BLIF, který je již standardním formátem v této problematice a lze v něm velice snadno popsat vše potřebné.

Generátor bude vytvářet obvody v závislosti na vstupních parametrech, jako je počet vstupních a výstupních signálů reprezentujících obvod zvenku, tak i velké množství vnitřních parametrů. Nejvýraznější z nich je počet úrovní obvodu a možnost nastavit počet úrovní mezi vstupy a jednotlivými výstupy. Maximální délka signálů omezuje počet úrovní obvodu, který je signál schopen překonat. K signálům také patří parametr omezující jeho maximální větvení a druhý definující průměrné větvení.

Logický prvek je vždy ze základní množiny (INV, AND, NAND, OR, NOR, XOR, NXOR), jejichž poměrný výskyt lze také nastavit, stejně tak i maximální počet vstupů do prvku. Samozřejmostí je nastavení množství logických prvků. Generátor bude vytvářet kromě kombinačních obvodů i sekvenční. Tato volba bude zahrnovat množství klopných obvodů i počet úrovní obvodu mezi vstupem a výstupem. Bude se také sledovat míra rekonvergenčí vzniklá v obvodu.

Vytvořené obvody budou složité k testování syntézních nástrojů, proto je nutné zabezpečit, aby výstupní signály nevedly na tautologii nebo kontradikci. Tím se zamezí úplné minimalizaci na pouhý konstantní signál. Obvody budou podrobeny minimalizaci a budou zjištěny závislosti parametrů na míru zmenšení obvodu.

Vytvoří se klony testovacích obvodů a budou porovnány jejich vlastnosti s originálem. Srovnání se provede jak na základě parametrů, tak i na míře rekonvergence. Hlavním ukazatelem, který bude sledován, je schopnost vytvořených obvodů dosáhnout obdobného stupně minimalizace.

Kapitola 2

Popis problému a specifikace cíle

Generátor logických obvodů bude, jak název napovídá, vytvářet logické obvody, které se co nejlépe přiblíží k zadaným parametrům. Stavebním prvkem obvodu je logické hradlo.

V této práci je jméno signálu (fyzického vodiče spojující jednotlivé logické prvky) totožné s pojmenováním hradla, ve kterém se signál vytváří. Pokud bude tento systém v některém okamžiku matoucí, přesněji dotyčný problém rozvedu.

2.1 Logická hradla

Základní typy hradel jsou popsány v tabulce 2.1. Jedná se o základní množinu, kterou budu používat pro generování, žádná jiná hradla se zde neobjeví. Důležitá informace v tabulce je o počtu vstupů hradel. Invertor má jeden, XOR a NXOR dva a ostatní hradla mají dva a více vstupů. Omezení u invertoru je logické, protože tento prvek pouze invertuje hodnotu signálu. Pouze dva vstupy u logického prvku XOR a NXOR jsou způsobeny tím, že ve fyzickém návrhu je sice možné vytvořit z tranzistorů tyto prvky s více než dvěma vstupy ale nevyplatí se to, protože jsou dodatečné vstupy vytvořeny pomocí kaskády dvouvstupých hradel. Z tohoto důvodu se v této práci budu zabývat pouze dvouvstupými prvky těchto typů.

Název	Zkratka	Logická operace	Počet vstupu
Invertor	INV	$y = \bar{a}$	1
Logický součin	AND	$y = a \cdot b$	2+
Negovaný logický součin	NAND	$y = \overline{a \cdot b}$	2+
Logický součet	OR	$y = a + b$	2+
Negovaný logická součet	NOR	$y = \overline{a + b}$	2+
Exkluzivní logický součet	XOR	$y = a \oplus b$	2
Exkluzivní logický součin	NXOR	$y = a \otimes b$	2

Tabulka 2.1: Tabulka typů hradel

Pravděpodobnost výskytu hradel bude možno nastavit. Toho se využije v druhé části diplomové práce. Ta se bude zabývat minimalizací vygenerovaných obvodů. Další parametry, podle kterých se bude obvod generovat jsou následující.

2.2 Počet vstupních portů

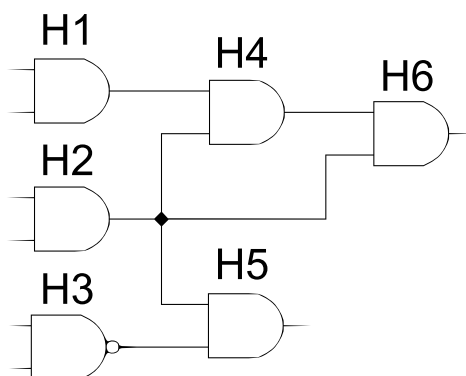
Jedná se o počet signálů vstupujících do obvodu.

2.3 Počet výstupních portů

Je to počet signálů, které vystupují z obvodu. Každý výstupní signál je závislý na určité množině vstupních signálů, tyto signály ovlivňují jeho hodnotu. Součet množin závislých vstupů všech výstupních signálů musí dát kompletní množinu vstupních signálů. Tím se zabezpečí, aby každý vstupní signál měl na obvod vliv. Tuto množinu vstupních signálů na kterých je hradlo závislé budu nazývat **pokrytí**.

Druhá podmínka souvisí s celou strukturou obvodu. Obvod se nesmí rozpadnout na dva nebo více samostatných obvodů. Jinak řečeno: pokrytí výstupního signálu nebo skupin signálů nesmí být disjunktní vůči jinému výstupnímu signálu nebo skupině. Uvedeno na příkladu: existuje obvod o čtyřech vstupech (a, b, c, d) a čtyřech výstupech (v, x, y, z). Pokud signály v a x budou závislé na a, b a zároveň y, z na c a d . Dojde k rozdělení na dva samostatné obvody. Tomu se musí zabránit.

2.4 Počet úrovní obvodu

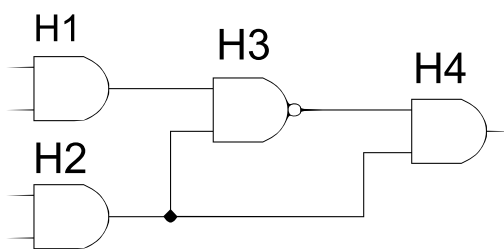


Obrázek 2.1: Ukázka úrovní obvodu

Počet úrovní obvodu je stanoven jako maximální počet hradel, skrz které projde signál od vstupu na výstup. Na obrázku 2.1 je obvod se šesti vstupy a dvěma výstupy. Je zde skupina hradel H1, H2 a H3, která náleží první úrovni. H4 se nachází na úrovni dvě a H6 na úrovni tři. Hradlo H5 se nachází v úrovni dvě, protože oba jeho předchůdci se nacházejí na úrovni jedna. Hradlo se tedy nachází v takové úrovni, která je o jednu větší než maximální úroveň hradel na jeho vstupech.

2.5 Maximální návaznost signálů

Jedná se o navázání výstupního signálu hradla na vstup jiného. To se nachází o určitý počet úrovní dále. Rozdíl úrovní je pojem **návaznost**.



Obrázek 2.2: Ukázka návaznosti signálů

Na obrázku 2.2 je znázorněn jednoduchý obvod. Ten se skládá ze čtyř logických hradel H1 až H4. Signál výstupu hradla H2 jde na vstup hradla H3, které je o jednu úroveň dále. Mezi H2 a H3 se jedná o návaznost rovno jedné. Signál H2 jde ale také na vstup hradla H4, mezi těmito hradly je rozdíl dvou úrovní, což je také největší návaznost, kterou lze v tomto obvodu nalézt.

2.6 Celkový počet hradel obvodu

Počet všech logických hradel nacházejících se v obvodu. Na toto číslo nemají vliv vstupní a výstupní porty a také počet klopných obvodů v případě generování sekvenčního obvodu.

2.7 Maximální počet vstupů hradla

Parametrem, který lze nastavit, je také maximální počet vstupů do hradla. Tato hodnota se pohybuje od čísla 2, pro striktně dvouvstupová hradla výše. Opět se zde pohybujeme v omezeních danými tabulkou hradel 2.1, proto nastavení třech vstupů a více se může týkat pouze logického součtu, součinu a jejich negací.

2.8 Maximální a průměrné větvení signálu

Každý signál, který se vyskytuje v obvodu, je buď vstupní nebo se jedná o výstup z hradla. Takový signál se může šířit do více hradel a tím vzniká větvení. Na obrázku 2.2 má výstupní signál hradla H2 větvení o hodnotě dva, na rozdíl od všech ostatních signálů, které mají větvení rovno jedné. Výstupní signál může být větvený, ale ne za účelem navýšení počtu výstupních signálů. Druhá větev signálů může vzniknout pouze jako vstup do jiného hradla. **Průměrné větvení** p_v je definováno jako:

$$p_v = \frac{\sum_{i=0}^n S_{v_i}}{n} \quad (2.1)$$

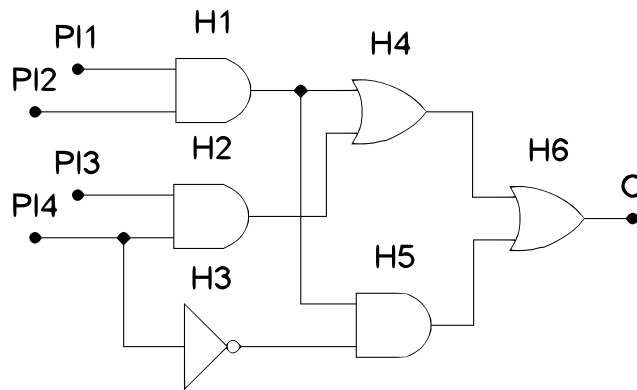
V čitateli je součet všech větvení signálů S_v a ve jmenovateli počet signálů n .

2.9 Počet úrovní obvodu mezi vstupy a výstupem

Dalším parametrem, který se bude nastavovat v této práci, jsou úrovně, z nichž budou vystupovat signály. Jedná se o číslo, které určuje z jaké úrovně bude onen výstup pocházet. Z toho vyplývá, že pokud je zapotřebí generovat obvod o celkovém počtu úrovní L , bude tato hodnota x dosahovat hodnot $x \leq L$. A minimálně jeden výstup z obvodu musí splňovat $x = L$, jinak by hodnota o počtu úrovní nebyla splněna.

2.10 Rekonvergence

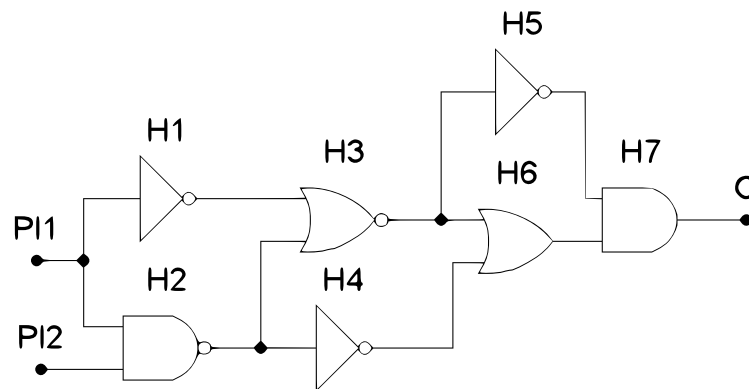
Rekonvergence vzniká při větvení signálu. Ten se rozdělí, jeho vliv se šíří a pokud se signály opět střetnou, vzniká rekonvergence.



Obrázek 2.3: Ukázka rekonvergence

Jak vypadá rekonvergence je zobrazeno na obrázku 2.3. Obvod se skládá ze čtyř vstupních signálů PI1 až PI4 a jedním výstupním O. V obvodu je zapojeno šest logických hradel H1 až H6. Nacházejí se zde dvě rekonvergence. První vzniká rozvětvením signálu PI4. Horní větev prochází hradly H2, H4 až k výstupnímu hradlu H6. Dolní větev sleduje cestu přes H3, H5 a opět až k H6, kde se spojuje s horní větví a vzniká rekonvergence, která celkově prochází přes tři úrovně obvodu. V obvodu 2.3 se nachází i další větvení, z něhož vzniká rekonvergence. Výstup z hradla H1 se větví a signál separátně prochází hradly H4 a H5, aby se hned spojil opět ve výstupním hradlu H6. Tato rekonvergence prochází dvěma úrovněmi. Dvoje větvení, z kterých vznikají dvě rekonvergence. Mohlo by se zdát, že z každého větvení vznikne nová rekonvergence, ale není tomu tak. Větvení signálu je podmínka nutná, nikoliv dostačující. Jednotlivé větve se mohou rozdělit a každá směřovat k jinému výstupu.

Na obrázku 2.4 je obvod se sedmi hradly, dvěma vstupy a jedním výstupem. Na první pohled je zřejmé, že obsahuje hned několik rekonverzí, konkrétně šest. První, druhá a třetí mají počátek ve větvení vstupního signálu PI1 a konce jsou v hradlech H3, H6 a H7. Čtvrtá a pátá začínají v signálu na výstupu H2 a konce jsou v H6 a H7. Šestá rekonvergence vzniká z větvení signálu H3 a končí v posledním hradle obvodu H7. Tento jev je způsoben větvením uvnitř smyčky v první rekonvergenci a následném spojení těchto větví. Pokud by takto obvod dále pokračoval, vznikalo by velké množství různě dlouhých rekonverzí, při přidání hradla AND se vstupy H5 a H7, by vznikla další smyčka



Obrázek 2.4: Ukázka obvodu se smyčkami rekonvergence

a celkový počet rekonvergenčí by narostl na hodnotu 10, což již přesahuje počet hradel v celkem jednoduchém obvodu.

V této práci se zaměřím na hledání, počítání a měření délky základních smyček rekonvergenčí, které jsou základním kamenem celého počtu rekonvergenčí, ale neprojeví se zde jev navyšování celkového počtu rekonvergenčí při sdružení smyček. V obvodu na obrázku 2.4 je hledaný počet tři. První vzniká větvením PI1 a ukončuje se v hradle H3, druhá má počátek v H2 a konec v H6, poslední je H3 s ukončením H7.

2.11 Sekvenční obvody

Parametrický generátor bude také moci vytvářet sekvenční obvody. Parametry, které se budou nastavovat jako u kombinačních obvodů, pouze zde přibude možnost určit počet klopných hradel typu D a počet úrovní kombinační logiky mezi vstupem a výstupem každého klopného hradla. Počet úrovní je obdobný parametr jako u výstupů, opět musí být maximální počet úrovní roven celkovému počtu úrovní obvodu.

2.12 Ostatní podmínky

Další důležitou podmínkou je vytvářet obvody tak, aby výsledný výstup nevedl na tautologii nebo na kontradikci. Konečné obvody budou poté analyzovat a minimalizovat nástroji na syntézu, proto je nutné zabezpečit, aby obvody nešly zaměnit za konstantní signály, vyvedené na výstupy.

Kapitola 3

Analýza problémů a stanovení postupu

Generování obvodů je komplexní záležitost. Ta žádá mnoho rozhodnutí během návrhu. V této kapitole se zaměřím na důvody, které mě vedly k výsledné podobě práce a podrobně zmíním svá rozhodnutí.

3.1 Základní pohled na problematiku

Nejdůležitější volba při tvorbě systému, který má za cíl něco vytvořit, je určit si systém generování. Od tohoto rozhodnutí se bude odvíjet další práce a tvorba algoritmů. Jako u většiny prací se tato volba učiní ihned na začátku. V mém případě se jedná o několik možností. Každá z nich se dá vytvořit tak, že se hledá obvod splňující všechny parametry, které je nutno splnit, nebo pouze podstatnou část a obvod následně upravovat s ohledem na nesplněné parametry.

- **Pozvolné generování a rozšiřování od počátečního triviálního semínka**

Triviální semínko je v mém případě jedno logické hradlo nebo předdefinovaná minimální struktura hradel. To by se za pomoci určitých pravidel rozšiřovalo a rostlo. Tento princip by si žádal, díky velkému počtu parametrů, množství pravidel pro generování a s tím spojenou náročnost výpočtu.

- **Genetický algoritmus**

Genetický algoritmus představuje zcela odlišný pohled na problematiku. Jednalo by se o řízenou populaci a křížení mezi jedinci. Tato metoda přináší více otázek než odpovědí. Problém by byl s určením fitness funkce i samotného genetického kódu jedince.

Tento postup zvolil můj kolega Tomáš Měchura [3] při vytváření obdobné práce na generování náhodných booleovských funkcí, konečných automatů a sítí hradel.

- **Generování velkého počtu a následný výběr**

Poslední možností je generování velkého počtu hradel, ze kterých by se následně vybral hledaný obvod.

Zvolil jsem poslední možnost. Generování velkého počtu hradel a z nich si budu vybírat hledaný obvod. Rozhodnutí však není černobílé, každá volba sebou nese rizika a budoucí problémy, které nejsou na první pohled znát. Já si však myslím, že jsem zvolil správně.

3.2 Generování hradla

Parametry základního elementu obvodu se generují v tomto pořadí:

1. Typ hradla

Nejdříve se náhodně vygeneruje, o jaké logické hradlo se bude jednat, z možností popsané v tabulce 2.1. Pro každý typ hradla lze uživatelsky stanovit pravděpodobnostní míru jejich výskytu. Pokud tak není určeno, pravděpodobnost je u všech hradel stejná.

2. Počet vstupních signálů

Provede se pouze u logického součtu, součinu a jejich negací, protože u ostatních typů je počet vstupů již specifikován. Výsledný počet je shora omezen vstupním parametrem.

Generování se provádí v tomto pořadí, protože je nutné zabezpečit poměrné rozdělení typů hradel. V opačném pořadí by toto nebylo možné.

3.2.1 Výpočet říditelnosti

Po vytvoření hradla a přidělení příslušných vstupů se musí zabezpečit, aby právě přidané hradlo nevedlo na tautologii nebo kontradikci. K tomu využijí základní parametr testovatelnosti obvodu tzv. výpočet říditelnosti C (viz. seznam použité literatury [1]). Hodnota této veličiny se pohybuje v množině $\langle 0, 1 \rangle$, což je pravděpodobnost, že signál nabývá hodnoty 1. Algoritmus je následující:

1. Pro všechny vstupní signály nastavím hodnotu C na 1.

2. Pro ostatní signály nastavím hodnotu C na ∞ .

3. Propaguji hodnoty C k novému hradlu.

Tato propagace je soubor jednoduchých rovnic, které se liší podle typu hradla. Výsledná říditelnost se značí $C(Y)$ a vstupní hodnoty jsou znázorněny $C(i)$ nebo konkrétně $C(A)$ a $C(B)$. Pravidla jsou následující:

- Invertor INV

$$C(Y) = 1 - C(A) \quad (3.1)$$

- Logický součin AND

$$C(Y) = \prod_{i=0}^n C(i) \quad (3.2)$$

- Logický součet OR

$$C(Y) = 1 - \prod_{i=0}^n (1 - C(i)) \quad (3.3)$$

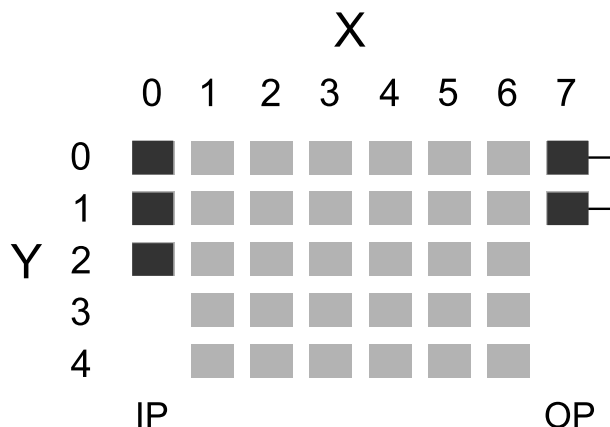
- Exkluzivní logický součet XOR

$$C(Y) = C(A) \cdot (1 - C(B)) + C(B) \cdot (1 - C(A)) \quad (3.4)$$

Výsledky této metody nemusí být, díky rekonvergenčním v obvodu, přesné, ale pro tento účel dostačující. Nově přidané hradlo musí splňovat $C(Y) < 90\%$ a zároveň $C(Y) > 10\%$. Tyto hodnoty jsou vhodné, protože nejsou přehnaně omezující, čímž by docházelo k velkému počtu nepřijatých hradel. Nejsou příliš mírné, tím by se ztráta hradel zminimalizovala. Díky nepřesnosti této metody, jako je citlivost na rekonvergenze, by vše ztratilo smysl.

3.3 Generování mřížky hradel

Zvolenou metodou je generování velkého počtu hradel, která se vkládají do mřížky, což je velice vhodné uspořádání, protože se již podobá výsledné struktuře obvodu. Počet úrovní obvodu lze zabezpečit velikostí mřížky a po dosažení konečné úrovně lze již vygenerovat jen takový počet hradel, kolik je výstupních signálů.



Obrázek 3.1: Struktura generování obvodu

Na obrázku 3.1 je zobrazena základní struktura. Obdélníky reprezentují hradla, která se do mřížky vkládají. Osa X reprezentuje úroveň obvodu. Osa Y jeho maximální šířku. Tou se budu podrobněji zabývat v následující kapitole 3.4. V první úrovni $X = 0$ se nacházejí reprezentace vstupních signálů. Mají podobu obdobnou jako ostatní hradla, aby bylo docíleno jejich rovnoprávného přidání do mřížky. Na pozici $X = 7$ jsou již jen dvě hradla, která reprezentují vstupní parametr na dva výstupní signály.

Hradla se generují postupně po úrovních a zapojují se vstupní signály, které jsou reprezentovány jinými hradly v mřížce. Vstupní signály se musí pokusit splnit podmínky. Pro každý první vstup platí:

- Vstupní signál/hradlo pochází z předešlé úrovně.
Tato podmínka zabezpečí správný počet úrovní obvodu.
- Pokud existuje takový vstup, který rozšíří pokrytí úrovně, použije se.
Úroveň obvodu jako celek musí být závislý na všech vstupních signálech. Tím se zabezpečí i závislost výstupních signálů na vstupních neboli pokrytí.

Pro zbylé vstupy platí následující:

- Vstup bude signál/hradlo, které se nachází v předchozích úrovních, které jsou omezeny návazností hradla. Vybírá se z množiny signálů o úrovních $\langle \text{aktualniUroven} - \text{navaznost}, \text{aktualniUroven} - 1 \rangle$
Hradlo je již pomocí prvního vstupu usazeno na požadované úrovni. Ta se shoduje s pozicí v mřížce. Ostatní vstupy mají volnost výběru signálů z více úrovní, které jsou charakterizovány vstupním parametrem návazností.

- Použije se takový signál, který rozšíří pokrytí úrovně.
 - Pokud neexistuje nebo je pokrytí úrovně již kompletní, použije se takový vstup, který rozšíří pokrytí hradla.
 - I další vstupy se musí pokusit o navázání na co největší množinu vstupních signálů.*
 - * Pokud takový neexistuje, nebo je pokrytí hradla kompletní, použije se libovolný dosažitelný z hlediska návaznosti.

Výsledný seznam vstupů se pouze vloží do právě vygenerovaného hradla. Každé hradlo ví pouze o svých předchůdcích, nikoli však následovnicích. To umožňuje při výběru jakéhokoli hradla procházet, či jinak pracovat s jeho předchůdci, aniž by se musela měnit struktura.

Generování mřížky končí vytvořením poslední úrovně, kde se nachází jen tolik hradel, kolik je počet zadaných výstupů. Dále se výsledný obvod jednoduše vyčlení z mřížky rekurzivním průchodem do hloubky, který se spustí postupně na všechna výstupní hradla. Tím se pojmenují všechna hradla obvodu a odliší se od nevyužitých. Při tomto algoritmu se také vytvoří dopředné vazby, kdy každé hradlo bude znát kromě svého předchůdce i následovníka. To poté zlepší práci v dalších algoritmech. Obvod je vytvořen.

Díky tomuto postupu se podařilo zabezpečit splnění několika parametrů. Počet vstupních signálů je zajištěn vložením vstupů do nulté úrovně mřížky hradel. Počet výstupů je určen hradly v nejvyšší úrovni podle zadaného počtu úrovní obvodu. Pravidla na vytváření hradla a jeho vstupů zabezpečují, že výstupní signály jsou navázané na všechny vstupní a také zabezpečují zadanou maximální návaznost signálů.

Jedná se o čistě náhodný proces, proto je zřejmé, že touto metodou nemůžeme nijak zabezpečit přesný počet hradel v obvodu. Dalším parametrem, který není touto metodou nastaven je počet úrovní mezi vstupy a výstupem. Tento způsob generování je základní a bude se dále rozšiřovat.

3.4 Přesný počet hradel

V této podkapitole si určíme a rozebereme způsob generování přesného počtu hradel. Díky náhodnému generování základní mřížky, nezbyvá již mnoho atributů, kterými by teoreticky šel nastavovat. Jediný možný nenastavený atribut, je šířka mřížky. Na obrázku 3.1 je označena jako Y . Šířka určuje kolik hradel se vygeneruje na každé úrovni mřížky. Takto vytvořená úroveň se musí minimálně jedním vstupem navázat na předchozí. Generování končí ve chvíli, kdy se celá mřížka zaplní, poté se oddělí využitá a nevyužitá hradla podle jejich závislosti na výstupech.

Nyní bude nutné nalézt závislosti mezi šířkou mřížky a výsledným počtem hradel.

Protože celý proces bude ovlivňovat pouze nastavování jedné proměnné, je důležité stanovit co nejmenší požadavky na hledání. Mezi přemrštěné lze pokládat hledání obvodu s přesným počtem výstupů na předem daných úrovních, což je také jeden z generujících parametrů. Problém se musí co nejvíce zjednodušit. Bude se tedy generovat menší počet úrovní obvodu a poté se vyhledají signály, které na sebe váží správný počet hradel.

Počet vázaných hradel se získá prohledáním do hloubky a sečtením hradel na cestě. To už samo o sobě má časovou náročnost $\Theta(h)$, kde h je počet nalezených závislých hradel. Počet hradel navazujících na jedno je:

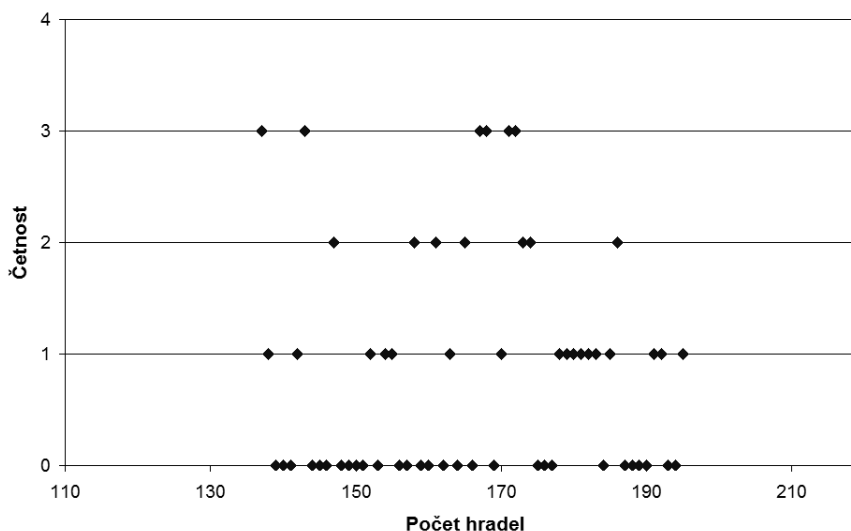
$$H = O\left(\sum_{i=0}^{l-1} \min(S, 2^i)\right) \quad (3.5)$$

Zde S udává šířku mřížky. Do celkového počtu se také započítá prvotní hradlo, jenž je na úrovni l . Jedná se o nejhorší možnost při generování. Všechna hradla musejí mít dva vstupy. Ve skutečnosti se tato hodnota při generování mění podle zadaného parametru. Každé hradlo by muselo na sebe vázat unikátní a nepoužité hradlo z minulé úrovně, čímž by se začal generovat úplný binární strom. Vše je velice nepravděpodobné.

Na grafu 3.2 je zobrazen histogram zobrazující počty hradel, které na sebe váží jednotlivé signály ve dvacáté úrovni při parametrech 3.1 a stejnou pravděpodobností pro výběr každého hradla.

Počet vstupů	4
Návaznost signálů	1
Maximální počet vstupu hradla	2
Šířka mřížky	50

Tabulka 3.1: Tabulka parametrů



Obrázek 3.2: Četnost počtu hradel na úrovni

Z grafu 3.2 je vidět, že teoretického maxima se ani zdaleka nedosahuje. Maximální hodnota je 195, kdežto vypočítaná hodnota podle vzorce 3.5 je $\sum_{i=0}^{20-1} \min(50, 2^i) = 763$

Tohoto jevu se využije následovně: vygeneruje se pouze jednovýstupový obvod, který bude později doupraven. Onen jediný výstup se vytvoří tak, že se vygenerují hradla do úrovně o jednu menší, než hledaný počet úrovní, a poté se naleznou dva signály, které na sebe váží hledaný počet hra-

del. Tyto se spojí v poslední úrovni dvouvstupovým hradlem. Tím vznikne jeden výstup na správné úrovni. Při hledání dvou specifických signálů je nutné projít tyto možnosti.

$$\begin{aligned}
 E &= \{e_1, e_2, \dots, e_S\}, |E| = S \\
 F &= \{e_1, e_2, \dots, e_S, \dots, e_{S \cdot N}\}, |F| = S \cdot N \\
 X &= [e_i, e_j] \\
 e_i &\neq e_j \wedge e_i \in E \wedge e_j \in F \\
 p &= S \cdot S \cdot (N - 1) + \binom{S}{2} \tag{3.6}
 \end{aligned}$$

Existují dvě množiny, pro každý vstup jedna. První E je tvořena hradly, které se nacházejí v nejvyšší úrovni. To je z důvodu dodržení úrovně posoupnosti. Druhá množina F je nadmnožinou E $F \supseteq E$. Obsahuje všechny dosažitelné signály. Symbol e_i znázorňuje jeden element množiny, v našem případě jedno hradlo/signál. S je šířka mřížky a N je maximální návaznost signálů. Jako předpoklad je brán fakt, že návaznost může být maximálně tak velká jako počet úrovní hledaného obvodu. Hledají se všechny kombinace dvou prvků $X = [e_i, e_j]$ pro které platí $e_i \neq e_j \wedge e_i \in E \wedge e_j \in F$. Nejdříve se vypočítá počet kombinací E s $F - E$, velikost $|E| = S$ a $|F - E| = S \cdot (N - 1)$, což dohromady dá $S \cdot S \cdot (N - 1)$, k tomu je nutné přičíst počet kombinací při hledání dvou prvků bez opakování z množiny E , tedy $\binom{S}{2}$.

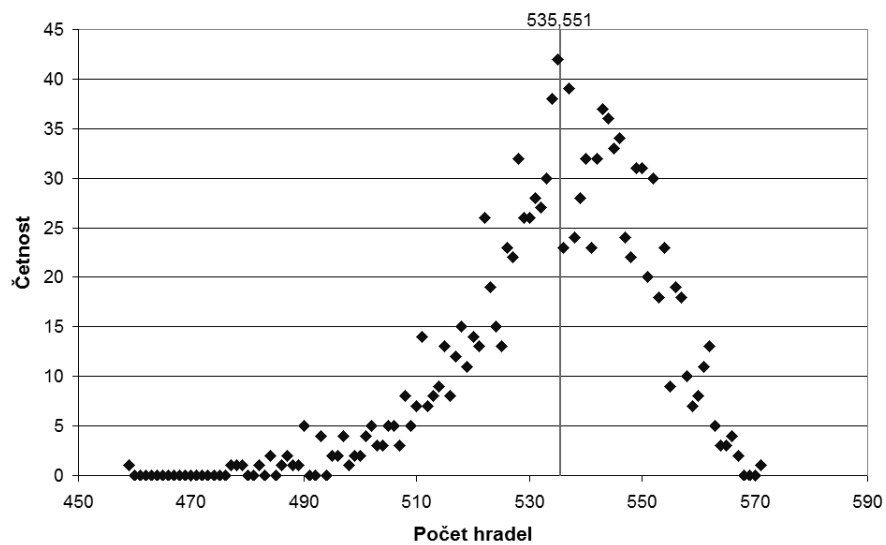
Počet možností je velký, ale není nutné je všechny projít. Stačí nalézt jednu, která odpovídá hledanému počtu hradel. Pokud se ale hledaný počet nenalezl, není možné obvod vytvořit. To se může stát i v případě, kdy je mřížka správně nastavena, ale počet možností je velice malý. Díky malé návaznosti signálů a malé šířce mřížky nedojde k nalezení hledané struktury. V tomto případě je zapotřebí program spustit znovu.

Na grafu 3.3 je zobrazena situace velikosti součtu množin dosažitelných hradel dvou signálů se stejnými parametry jako v minulém případě 3.1. Na ose X je počet hradel a na ose Y se nachází jejich četnost. Počet nalezených hodnot při návaznosti signálů pouze z minulé úrovně je roven kombinaci $\binom{S}{2} = \binom{50}{2} = 1225$. Na grafu je vyznačen průměr 535,551. Nachází se v blízkosti nejčetnějších hodnot, pokud se bude tedy hledat takový počet hradel, který bude blízko tohoto průměru, je velká šance na správnou kombinaci signálů a tudíž hledání nalezne rychle správný výsledek.

	#1	#2	#3	#4	#5	#6	Průměr	Min	Max
M1	2666,5	2695,3	2656,4	2692,8	2726,0	2698,4	2689,2	2656,4	2726,0
M2	539,6	555,9	537,9	537,1	538,0	552,9	543,6	537,1	555,9

Tabulka 3.2: Tabulka naměřených průměrů

V tabulce 3.2 se nachází rozptyl průměrných hodnot při opakování měření, obvod se generoval s parametry zobrazené v tabulce 3.3. Při měření $M1$ a $M2$ se hodnoty průměru pohybují v rozsahu $\pm 3\%$ od zprůměrovaných hodnot. Jedná se o malé rozdíly, což dokazuje, že průměrná hodnota počtu hradel, které navazují na dva signály je stabilní, proto budu dále pracovat s hodnotou průměru a ne s rozsahem počtu navázaných hradel.



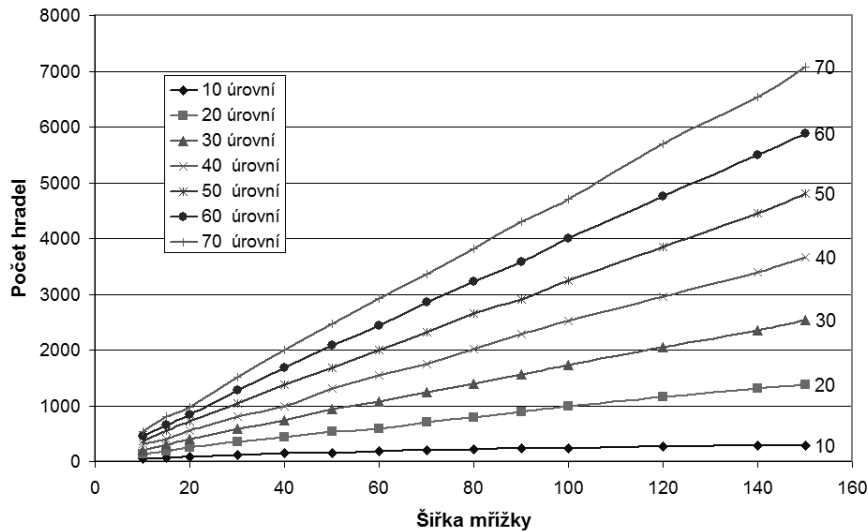
Obrázek 3.3: Četnost počtu hradel pro dva vstupy

	Počet vstupů	Návaznost	Počet úrovní	Max počet vstupů hradla	Šířka mřížky
M1	5	1	40	3	100
M2	5	1	20	2	50

Tabulka 3.3: Tabulka naměřených průměrů, parametry měření

Pro vyhledání přesného počtu hradel bude nutné nastavit šířku mřížky tak, aby se hledaná hodnota co nejvíce přiblížila průměru. V následujících podkapitolách se budu zabývat vlastnostmi mřížky a na konci se pokusím vytvořit postup, který nastaví šířku podle počtu hradel šířku mřížky.

3.4.1 Závislost počtu hradel na šířce mřížky



Obrázek 3.4: Závislost počtu hradel na šířce mřížky

Na obrázku 3.4 je graf, který znázorňuje změny průměrného počtu navazujících hradel na dva signály. Parametry tohoto měření jsou v tabulce A.2. Návaznost signálů je omezena na rozdíl jedné úrovně. Počet vstupních signálů je čtyři a co se týče hradel, jejich počet vstupních signálů je shora omezen dvěma. Naměřená data se nacházejí v příloze A.1. Hledané obvody mají počet úrovní od 10 do 70, to znamená, že dvojice signálů se hledá na úrovni o jedna menší.

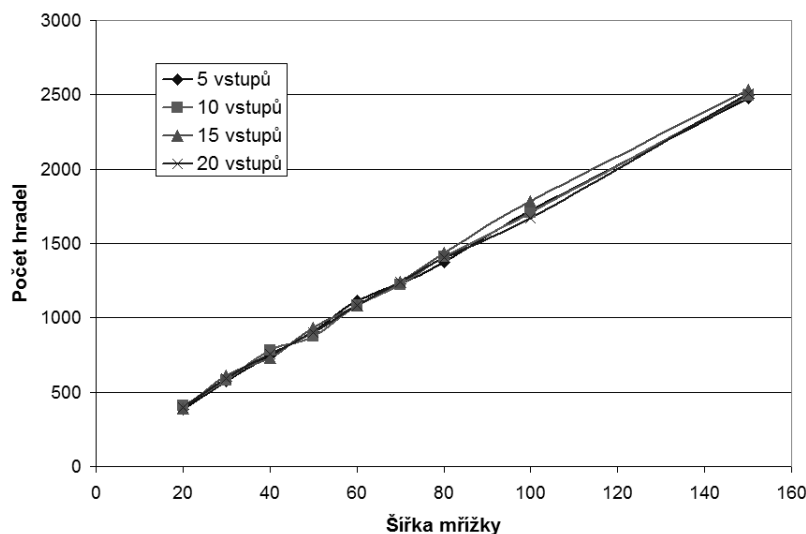
Z grafu je zřejmá lineární závislost, která jakoby vychází ze společného počátku. Toto zjištění je velice důležité, protože lze se dvěma body na charakteristice vypočítat jakýkoli bod.

3.4.2 Závislost počtu hradel na počtu vstupních signálů

Počet vstupních signálů je důležitý parametr z hlediska vytváření logických obvodů, ovšem zde jeho důležitost není tak výrazná. Důvod je zřejmý již z pravidel generování jednotlivých hradel, každé hradlo na sebe váže co nejvíce vstupních signálů nebo jeho následovníků. Toto pravidlo zapříčiní velmi rychlé provázání vstupů s hradly a ihned, jak je to možné, logický člen je již ovlivňován všemi vstupními signály, poté již nemá tento parametr žádný vliv.

Výsledná charakteristika je vidět na grafu 3.5. Jako výchozí parametr se nyní mění počet vstupů a konstantní zůstává počet úrovní obvodu na třiceti a maximální návaznost vstupních signálů hradla na jedné. Parametry jsou přehledně zobrazeny v tabulce A.4, stejně jako naměřená data A.3.

Na obrázku není vidět žádná závislost, proto můžeme tento parametr při nastavování mřížky ignorovat.



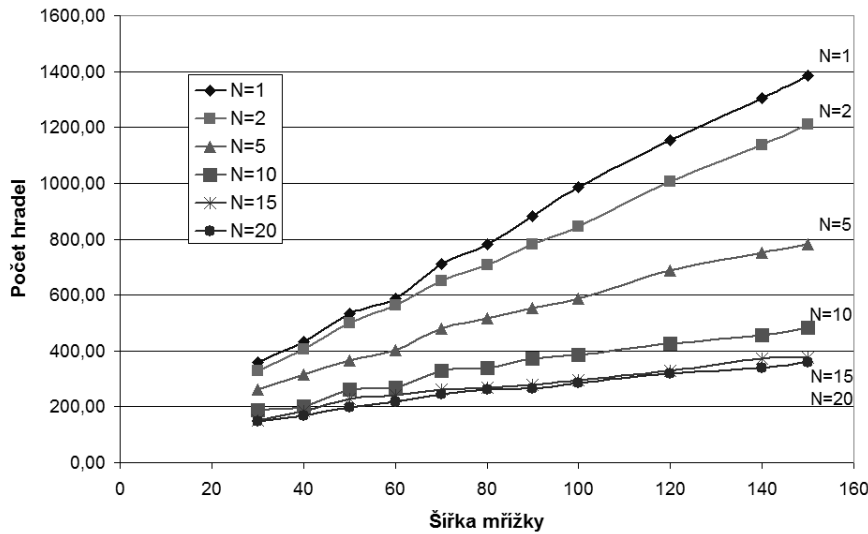
Obrázek 3.5: Závislost počtu hradel na počtu vstupů obvodu

3.4.3 Závislost na maximální návaznosti signálů

Omezení návaznosti signálů je velice ovlivňující parametr. Při striktním nastavení na hodnotu jedna musí každé logické hradlo navazovat všemi vstupy na signál z minulé úrovně, čímž dochází k velkému zvyšování počtu navazujících hradel. Pokud se parametr zvýší, bude docházet k úbytku hradel, protože u jednoho vstupu bude často docházet navázání signálu z nižších úrovní, tedy i s menším počtem navazujících hradel.

Graf 3.6 znázorňuje situaci, kdy se generuje obvod o dvaceti úrovních, pěti vstupech a s maximálně dvěma vstupy hradla. Parametry jsou přehledně v příloze A.6 spolu s tabulkou dat A.5. Návaznost se mění v rozmezí $\langle 1, 20 \rangle$ kde horní omezení je dáno právě celkovým počtem úrovní. Větší počet již nemá smysl a chová se stejně jako hodnota dvacet. Charakteristiky jednotlivých nastavení jsou opět lineární, ale se zvyšujícím se parametrem návaznosti sklon klesá. Pokles není rovnoměrný, zvýšení návaznosti signálů z hodnoty jedna na dva se projeví v průměrném počtu navazujících hradel prakticky stejně výrazně, jako změna z hodnoty deset na dvacet a změna z patnácti na dvacet je již skoro nepatrná.

Parametr maximální návaznosti vstupních signálů je další z parametrů, který ovlivňuje výslednou charakteristiku. Při nastavování šířky mřížky je nutno s ním počítat.



Obrázek 3.6: Závislost počtu hradel na návaznosti signálů

3.4.4 Závislost na maximálním počtu vstupů hradla

Poslední nastavení, které zbývá prověřit, je změna maximálního počtu vstupů hradla. Jak již víme z předchozích kapitol, tři logická hradla (*INV*, *XOR*, *NXOR*) ze sedmi mají již svůj počet vstupů pevně určen na hodnotě dva. Tento parametr se tedy týká ostatních hradel (*OR*, *NOR*, *AND*, *NAND*).

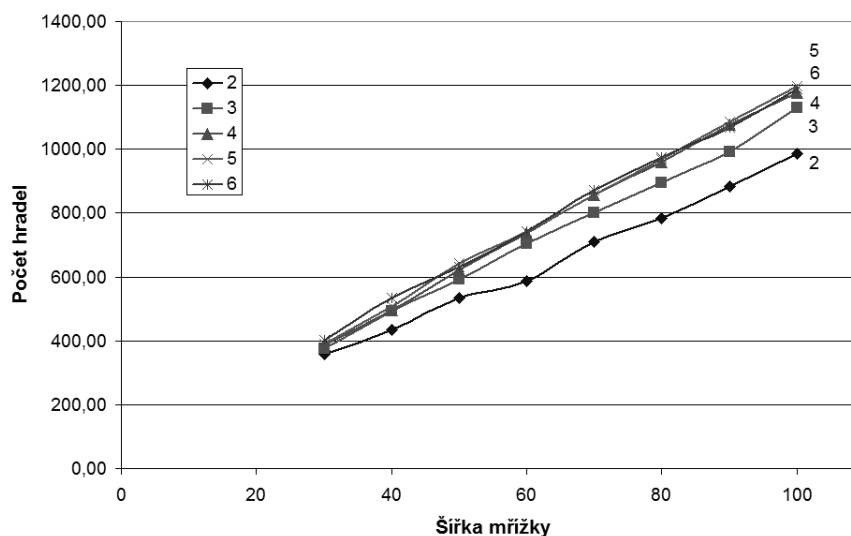
Jaká je pravděpodobnost, že se vytvoří hradlo se stanoveným maximem? Pravděpodobnost, že se vybere hradlo, které bude mít možnost většího počtu vstupů je tedy $\frac{4}{7}$. Toto se vynásobí šancí náhodného vybrání maximální hodnoty $\frac{1}{max-1}$. Výraz $max-1$, protože vylučujeme jeden vstup. Celková pravděpodobnost je:

$$P = \frac{4}{7} \cdot \frac{1}{max-1} \quad (3.7)$$

Výsledná charakteristika při zvyšování tohoto parametru bude stále strmější. Hradla na sebe budou vázat více předchůdců. Růst se zastaví ve chvíli, kdy pravděpodobnost dosažení maximálního parametru bude malá a již se neprojeví.

Graf 3.7 potvrzuje teoretický rozbor. Neměnné parametry a data jsou v příloze A.7 a A.8. Nejvíce se projeví změna parametru ze dvou na hodnotu tři a mezi hodnotami pět a šest již není žádný rozdíl.

Tento parametr příliš neovlivňuje generování obvodu, ale je nutné s ním počítat. Uživateli bude spíše preferovat nižší hodnoty.



Obrázek 3.7: Závislost počtu hradel na max počtu vstupů hradla

3.4.5 Heuristika nastavení šířky mřížky

Po projití všech důležitých parametrů již známe jejich závislosti ovlivňování počtu hradel v obvodu. Nyní je na řadě vytvořit model těchto návazností, který při startu programu nastaví šířku mřížky tak, aby hledaný obvod šel vygenerovat. Po dlouhém hledání a zkoušení různých matematických modelů jsem došel k následující metodě, která je velice přesná. Bude se jednat o externí soubor, kde se nachází předvypočítané hodnoty, s nimiž se bude následně pracovat. Jistě nejde přepočítat všechny možné kombinace nastavení, proto se tato metoda opře o lineární charakteristiky a v souboru bude jen prosté minimum hodnot nutných k výpočtu.

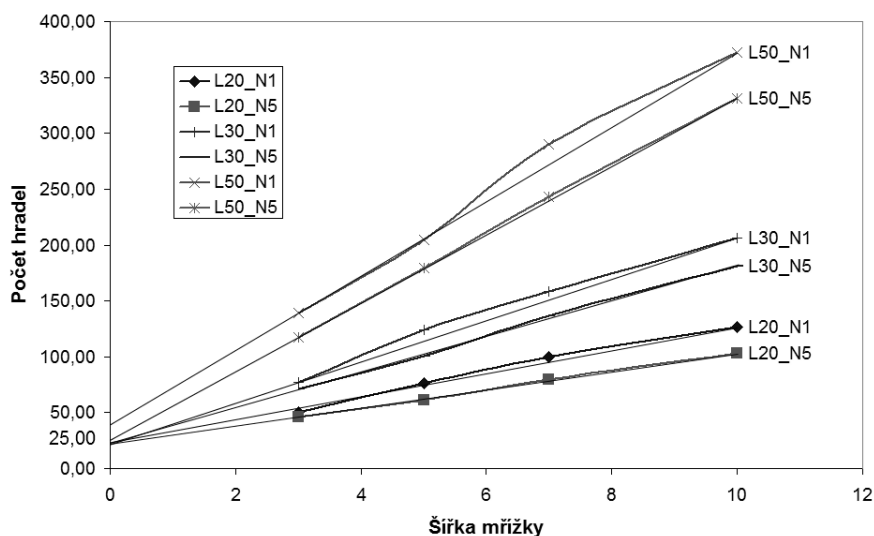
Z předešlých grafů 3.4, 3.5, 3.6 a 3.7 vyplývá tendence k neměřitelnému počátečnímu bodu všech charakteristik. Protáhnutím přímkou by se tento bod dal zjistit.

Na grafu 3.8 k protažení došlo. Použilo se zde šest různých závislostí, které se většinou spojily v bodě $[0, 25]$. Tento bod nemusí být absolutně přesný, protože jde o průměrnou hodnotu, kolem které je rozptýl reálných hodnot.

Druhý bod lineární charakteristiky by se měl nacházet co nejdále od počátečního, aby se co nejméně projevil možné tolerance generování. Pro druhý bod jsem zvolil šířku mřížky 100.

Výpočet bude proveden pro taková nastavení, která ovlivňují charakteristiku počtu hradel na šířce mřížky. Z předešlých podkapitol vyplývá počet úrovní obvodu a návaznost vstupních signálů jako nutné parametry. Maximální počet vstupních signálů hradla již není tak zřejmý.

V tabulce 3.4 jsou zobrazeny ideální šířky mřížky při změně tohoto parametru. Výpočet proběhl tak, že se napočítaly hodnoty pro tři různé úrovně obvodu (25, 30, 50) se stejnou návazností (5) pro šířku mřížky 100, poté se obdobně vytvořila charakteristická lineární funkce. Pomocí ní se vypočítaly šířky mřížky pro konkrétní počet hradel. Tyto hodnoty jsou v procentech vůči referenční hodnotě při dvou vstupech hradla. Nezáleží na výběru počtu hradel, pro všechny hodnoty vycházejí stejné poměry. To je dáno společným počátečním bodem.



Obrázek 3.8: Počáteční bod lineární závislosti

Max vstupy hradla	M1 [%]	M2 [%]	M3 [%]
2	100	100	100
3	85,92	87,69	90,49
4	78,93	82,32	86,24
5	77,82	82,05	85,23
6	78,61	82,94	84,63

Tabulka 3.4: Naměřené změny šířky mřížky v závislosti na maximálním počtu vstupu hradla

Z tabulky vyplývá, že je velice těžké určit nějaký koeficient, který by upravoval hledanou hodnotu. Pro pět vstupů do hradla je rozdíl mezi měřeními M1 a M3 7.41 %, což je znatelný rozdíl. Během vývoje heuristiky jsem pracoval s tímto poměrným systémem, ale čas ukázal, že není spolehlivý. Proto se musí přidat i tento parametr na změření přesných závislostí.

Měření bude probíhat na těchto hodnotách. Počet vstupu hradla: {2, 3, 4, 5}. Při dalším zvyšování tohoto parametru již nejsou prakticky žádné změny sklonu charakteristiky. Počet úrovní obvodu: {5, 6, 7, 8, 9, 10, 15, 20, ..., 100, 110, 120} a maximální možná návaznost vstupních signálů: {1, 2, 3, 4, 5, 10, 15, 20, ..., 100, 110, 120}. Tento parametr bude nastavován maximálně do hodnoty počtu úrovní. Hodnoty jsou rozestaveny v malých rozestupech rovnoměrně od pěti úrovní do sto dvaceti. Bude samozřejmě počítat s průměrnou hodnotou několika měření. Při hledání správné hodnoty se postupuje následovně.

- Určí se počet vstupů hradla, kde maximum je pět.
- Poté se vyhledá hodnota návaznosti. Pokud hodnota v tabulce není, použije se nejbližší menší.
- Vyhledá se y souřadnice pro hodnotu počtu úrovní. Pokud přesná hodnota není, lineárně se

aproximuje výsledná hodnota ze dvou okolních. Je-li je hledaná úroveň větší než maximální předpočítaná, provede se lineární extrapolace ze dvou nejvyšších hodnot.

- Po získání poslední souřadnice se z bodů $[0, 25]$, $[100, y]$ vytvoří hledaná přímka závislosti mezi průměrným počtem hradel a šířkou mřížky. Následně se vypočítá optimální šířka, se kterou se bude následné generování provádět.

$$H = k \cdot S + q$$

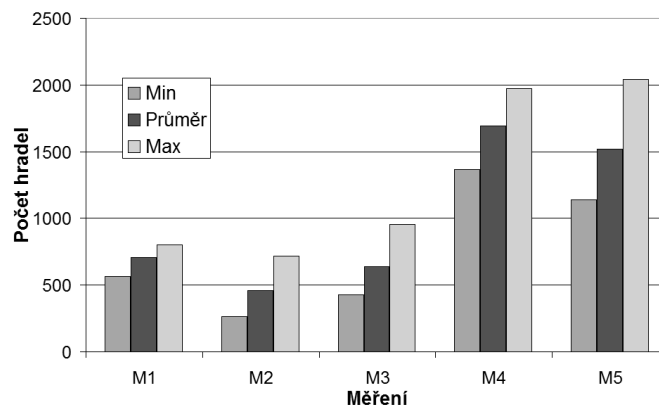
$$k = \frac{by - ay}{bx - ax} = \frac{by - 25}{100 - 0}$$

$$q = ay - k \cdot ax = 25$$

$$S = \frac{H - q}{k} = \frac{H - 25}{\frac{by - 25}{100}} = 100 \cdot \frac{H - 25}{by - 25} \quad (3.8)$$

Rovnice popisuje výpočet. H je průměrný počet hradel a S značí šířku mřížky. Koeficienty lineární rovnice jsou k a q .

Dalším faktorem, který byl do této chyby opomíjen, je skladba a preference jednotlivých hradel a tím i ovlivnění struktury generovaných obvodů, např.: velká preference invertorů atd. Tyto okolnosti se neberou v potaz, protože veškeré odchylky se ztratí v rozptylu počtu hradel.



Obrázek 3.9: Zobrazení rozptylu hodnot okolo průměrné hodnoty

Jak je tento rozptyl markantní, můžeme vidět na grafu 3.9, kde jsou zvýrazněny minimální a maximální hodnoty vůči průměru. Data s parametry jsou v příloze A.11. Všechna měření probíhala při stejné šířce mřížky (80), ale jinak se parametry rozcházejí. Měření $M1$ má nejmenší rozptyl hodnot, jedná se o dvacetúrovňový obvod s malou návazností (2), přesto je zde rozdíl mezi maximem a minimem 238, což odpovídá pohybu hodnot od -20 % do +14 % od průměrné hodnoty. U měření $M2$ je tento rozsah od -43 % do +56 %, $M3$ od -33 % do +50 %, $M4$ od -19 % do +17 % a $M5$ od -25 % do +34 %.

Je však nutné zdůraznit, že největší četnost hodnot se nalézá okolo průměru a množina možného počtu hradel neobsahuje všechny hodnoty mezi maximem a minimem.

Tímto jsme získali metodu a postupy jak generovat obvody s přesným počtem hradel. Bohužel se prozatím jedná o jednovýstupový obvod, proto se bude muset ještě dopracovat.

3.5 Vytvoření více výstupů

V předešlé kapitole 3.4 byl objasněn postup pro generování obvodu s přesným počtem hradel. Získáním tohoto důležitého parametru se ale docílilo ztrátou možnosti více výstupů obvodu. Je nutno dogenerovat zbylé výstupy tak, aby byl dodržen požadavek na počet logických hradel a také na určení počtu úrovní obvodu mezi vstupy a specifickým výstupem. Musí se jednat o řešení, které toto umožní a zároveň bude předem znám přesný počet hradel pro doplnění výstupů. O toto množství se sníží hledaný počet hradel jednovýstupového obvodu a po dogenerování se dosáhne hledaného počtu.

Požadavky na hledaný systém jsou následující:

- Hledaná struktura nesmí být tvořena velkým počtem hradel, aby co nejméně ovlivňovala snižováním hledání jednovýstupového obvodu. Při generování relativně malého obvodu s více vstupy by jinak mohlo toto řešení zabránit úspěšnému generování.
- Struktura musí maximálně využívat délku návaznosti vstupních signálů hradel. Tento bod je důležitý proto, aby struktura přílišně nezatěžovala poslední úroveň obvodu. Dejme příklad obvodu. Hledáme obvod o dvaceti úrovních, osmi výstupech a s relativně málo hradly. Povedlo se nalézt jednovýstupový obvod s patřičným počtem, díky povolené vysoké návaznosti vstupních signálů, ale je velice řídký. K takovému obvodu je nutné vytvořit dalších sedm výstupů, které v nejhorším případě musejí vycházet z poslední dvacáté úrovně. Je nutné zabezpečit, aby přidaná struktura navazovala s co nejvíce signály na původní obvod.
- Na druhou stranu nesmí být až moc triviální. To by vedlo na tvoření duplikátů výstupů a ve velkých obvodech na snížení říditelnosti, respektive pravděpodobnosti, že některý z výstupů bude tautologie nebo kontradikce.

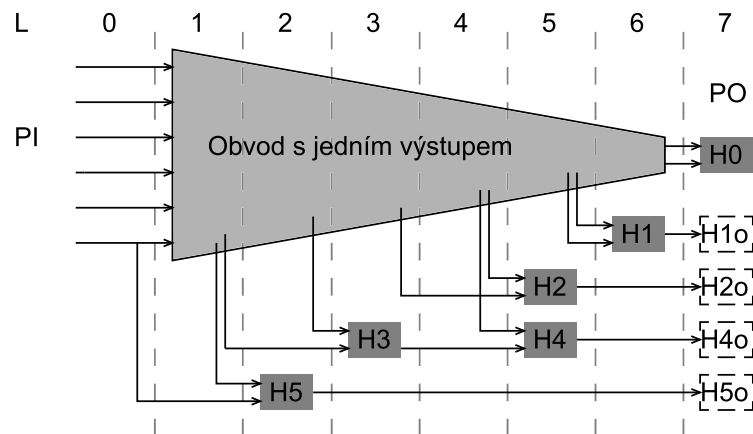
Všechna přidávaná hradla budou mít pouze dva vstupy. Tím se zjednoduší hledání struktury, protože tak odpadají různé eventualy.

Nepracuje se pouze s počtem výstupů, ale také s hodnotami, které udávají, kolik úrovní hradel se nachází mezi vstupy a jednotlivým výstupem. Tato množina je seřazena sestupně. V tomto pořadí se bude i generovat. První výstup je již hotový a je tvořen jedním dvouvstupovým hradlem. Následující výstup se vytvoří z počtu hradel o jednu větší, než minulý, ale jen pokud se jedná o výstup ze stejné úrovně. Pokud ne tento výstup bude vytvořen opět pouze jedním dvouvstupovým hradlem. Maximální počet hradel, které vytváří jeden vstup je následující:

$$\text{maxHradel}(x) = \lceil x / \text{maxNavaznost} \rceil \quad (3.9)$$

Hodnota x je úroveň aktuálního výstupu. Ve jmenovateli vzorce 3.9 je maxNavaznost , protože hradla na sebe navazují a jsou vždy v maximální vzdálenosti úrovní od sebe. Výjimku tvoří pouze

hradlo na nejnižší úrovni při dosažení maximálního počtu přidaných hradel na aktuální výstup. Úroveň tohoto hradla je dána maximální návazností. To zajistí, aby hradlo nebylo navázáno pouze na vstupy, pokud to není nutné v případě, kdy je návaznost rovna jedné. Každé takto vložené hradlo musí mít minimálně jeden vstup z hradla minulé úrovně. Tím se zabezpečí, že vkládané hradlo je na správné úrovni.



Obrázek 3.10: Vytvoření více výstupů

Na obrázku 3.10 je znázorněn výsledný princip. Obvod o sedmi úrovních, návazností vstupních signálů maximálně o dvě úrovně menších a s pěti výstupy, které mají vycházet z úrovní $\{7, 6; 5; 5; 2\}$. Hradlo H0 bylo vytvořeno již s jednovýstupovým obvodem, druhý výstup má pocházet z úrovně 6, což je jiná hodnota než minulý vstup. Proto se použije pouze jedno hradlo na stanovené úrovni H1. První vstup pochází u úrovně 5 původního obvodu a druhý může využít maximální návaznosti, tedy úrovní 5 a 6. U dalšího výstupu je situace obdobná. Výstup se nachází v úrovni 5. Změna přichází až s výstupem číslo 4, ten se vytváří na stejné úrovni a proto je tvořen dvěma hradly H3 na úrovni 3 a H4 na úrovni 5 a jsou propojené. Zbývá poslední výstup, kde se situace opakuje jako u výstupů číslo dva a tři.

Tato nově vložená hradla pocházejí z množiny $\{AND, NAND, OR, NOR, XOR, NXOR\}$ a jejich řiditelnost se musí udržet v přijatelných mezích.

Další hradla, která se objevují v obrázku 3.10 jsou H1o, H2o, H4o a H5o. Nejedná se přímo o hradla, ale o zástupce hradla na nižších úrovních. Jsou zde proto, aby se dodržel základní princip mřížky, při kterém jsou všechny výstupy na poslední úrovni.

Tento princip vytváření výstupů dává obvodům jednu vlastnost. Veškeré výstupní signály již dále nevcházejí do jiných hradel, což se v reálných obvodech neděje. Tento ústupek jsem volil z důvodů, že před generováním není možné znát přesnou strukturu obvodu, respektive počty signálů na jednotlivých úrovních jednovýstupového obvodu. Bez této znalosti není možné stanovit, kolik výstupů na určitých úrovních lze vyvést a kolik jich bude nutné vytvořit dodatečným přidáním hradel. Před samotným generováním je však nutné znát přesný počet hledaných hradel v jednovýstupovém obvodu z důvodů nastavení šířky mřížky a následného hledání.

3.6 Pojmenování signálů

V současné chvíli máme již základní obvod vytvořen, ale v mřížce se nyní nachází hradla, která vědí jen o svých předchůdcích a velká část z nich nepatří do výsledného obvodu. Jediné, co odděluje využitá hradla, je jejich dosažitelnost od některého z výstupů. Byl již nalezen obvod s hledaným počtem hradel a proto není nutné pracovat s nevyužitými hradly.

Procházením do hloubky zavolané postupně na výstupní hradla, se využité prvky unikátně pojmenují a také se naváží dopředné vazby, aby bylo možné procházet obvod všemi směry. Nyní nebrání nic k vytvoření druhé struktury, která bude obdobná původní mřížce a přesunout použitá hradla do ní.

V této podobě je již možné procházení obvodu nejen od vstupů k výstupům a opačně, ale také má každé hradlo svoji unikátní souřadnici, kde první hodnota určuje úroveň hradla a druhá její pořadí v ní.

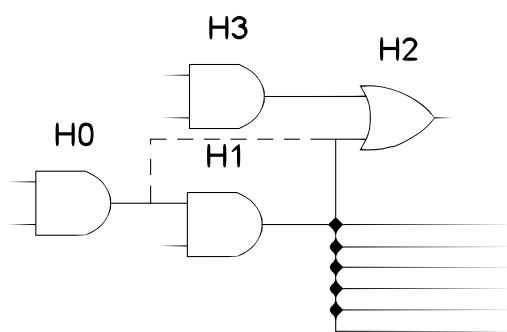
3.7 Snížení maximálního a průměrného větvení signálů

Obvod je již vygenerovaný, nyní se musí poupravit jeho parametry. Po každé upravě se nesmějí změnit již nastavené parametry, jako je počet úrovní obvodu, počet hradel, říditelnost výstupů ve stanovených mezích a všechny vstupní signály musí ovlivňovat výstup obvodu.

3.7.1 Maximální větvení signálu

Po přeuspořádání mřížky je možné sekvenčně všechna hradla projít a do nové struktury je seřadit v závislosti na počtu následovníků, což je hodnota větvení každého signálu.

Nyní budu brát hradlo/signál s největším větvením a pokusím se ho snížit. Budu procházet všechny následovníky a zkoušet, zda by nešlo zaměnit tento signál s velkým počtem větvení za nějaký s menším.



Obrázek 3.11: Snížení větvení

Na obrázku 3.11 je zobrazena klasická situace. Signál vycházející z hradla H1 se rozvětjuje do sedmi hradel. Důležitým parametrem v tomto algoritmu je návaznost signálů, která určuje rozsah možností pro změnu zapojení signálů, v tomto případě dva. Náhodným procházením větvení signálu

se algoritmus dostal na větev, která vchází do hradla H2. Procházením seřazené množiny od signálů s nejmenším větvením, se hledá vhodný signál, který by nahradil H1 na vstupu do hradla H2. Výsledné hradlo musí splňovat následující podmínky:

- Úroveň nového hradla musí být od změněného hradla maximálně o návaznost menší. Pokud je nahrazovaný signál jediný, který je z úrovně o jednu menší, musí být i nový z této úrovně.
- Po každé změně se musí provést přepočítání říditelností a výstupní signály musí mít tento parametr ve stanovených mezích.

Jako vhodné bylo vybráno hradlo H0. Jeho výstup má hodnotu větvení rovno jedné, nedochází k větvení. Hradlo H2 má již jeden vstup z minulé úrovně H3, proto můžeme použít toto nové hradlo, které je v dosahu návaznosti. Výsledek je jasný. Větvení H1 se snížilo na úkor větvení H0.

Pokud by se nepodařilo přeměrovat ani jednoho následovníka kritického signálu, je zde poslední možnost. Jedná se o smazání větvení. To lze jen pokud by následné hradlo, v našem příkladě H2, mělo více jak dva vstupy a existoval by ještě minimálně jeden vstup do takového hradla, který by měl původ v hradle o jednu úroveň menší. Musí se stále dodržet pravidlo, díky kterému má každé logické hradlo přesně definovanou úroveň.

3.7.2 Průměrné větvení signálu

Průměrné větvení signálu je komplikovaná záležitost. V rovnici 2.1 je popsáno, že výsledné průměrné větvení se vypočítá jako součet všech větvení na signálech děleno počtem signálů. Pokud si rovnici rozevíšme, vyplyne komplexnější závislost.

$$pv = \frac{\sum_{i=0}^n Sv}{n} = \frac{\sum_{i=0}^{Nh} Hi + PO}{Nh + PI} \quad (3.10)$$

$$PO \ll \sum_{i=0}^{Nh} Hi$$

$$PI \ll Nh$$

$$pv = \frac{\sum_{i=0}^{Nh} Hi}{Nh} \quad (3.11)$$

Každý signál, obsažený v obvodu musí někam vést a každý signál musí být vytvořen hradlem nebo přiveden jako vstup. V rovnici 3.10 je to popsáno. Součet všech signálů Sv se rovná součtu vstupů všech hradel Hi a počtu výstupních signálů PO . Ve jmenovateli se počet signálů n rovná počtu hradel Nh zvýšený počtem vstupních signálů PI .

Pokud předpokládáme, že počet vstupů PI je mnohem menší než počet hradel Nh , můžeme tento parametr zanedbat. Jedná se také o vstupní parametr generátoru, což můžeme brát za konstantu. Dále lze předpokládat, že počet výstupních signálů PI bude mnohem menší, než součet vstupů hradel $\sum Hi$, opět tento parametr můžeme zanedbat. Po úpravě vznikla rovnice 3.11, ze které vyplývá, že průměrný počet větvení se rovná součtu vstupů hradel dělený počtem logických hradel.

Výše zmíněné nenechává velký prostor na úpravu tohoto parametru. Snížení průměrného větvení signálů lze pouze snižováním počtu vstupů hradel, protože počet hradel je pevně dán. Minimální počet vstupních signálů je v tomto případě dva. Nelze tedy odstranit z dvouvstupového hradla *NAND* jeden vstup a tím vytvořit *INV*. Tento postup by vedl k velkému nárůstu invertorů v obvodu. Pokud se tedy generuje obvod s parametrem omezující počet vstupů na dva, nelze s průměrným větvením nic udělat. Pokud je parametr vyšší, lze snížit počet vstupů u hradel s větším počtem než dva. Nejdříve se vytvoří seznam takových hradel a poté se budou náhodně vybírat. U každého takto vybraného hradla se zkusí zrušit jeden jeho vstupní signál. Vstup hradla, a tím i jedno větvení signálu, se odstraňuje jednoduše, ale musí být dodržena tato pravidla:

- Vstupní signál musí obsahovat minimálně rozvětvení do dvou hradel. Pokud by tomu tak nebylo, signál by ztratil propojení s výstupem a vznikla by tím zbytečná větev obvodu. Také by se mohly přerušit vazby od některého ze vstupů k výstupu.
- Po změně se musí provést přepočítání říditelnosti výstupů a pokud by i jeden výstup neprošel nastavenými parametry, operace se vrátí zpět.

3.8 Hledání rekonvergenčí

Jak již bylo řečeno, rekonvergence je rozvětvení a následné sloučení signálu v obvodu.

Já se budu zabývat základními smyčkami rekonvergenčí a k jejich přesné definici potřebuji následující údaje:

- **Signál**
Základní parametr je signál, který se větví. Tím je specifikováno i hradlo místo vzniku signálu. To udává i úroveň, ve které se signál větví.
- **Větvení**
Další charakterizující vlastností rekonvergenčí je definice větvení, tedy následující hradla, kudy daný signál při rozvětvení procházel a v konečném důsledku vytváří onu rekonvergenčí.
- **Konečné hradlo**
Jedná se o hradlo, ve kterém vstupní signály vytvoří rekonvergenčí.

Díky těmto parametrům lze jednoduše zjistit počet úrovní, které jsou mezi počátečním větvením a následným spojením.

Algoritmus není příliš složitý. Bude se jednat o průchod obvodem od vstupů k výstupům a propagaci cest jednotlivých vstupních signálů. Každé hradlo má informaci o všech svých předchůdcích, rozděleným podle návaznosti na vstupních signálech. Při výpočtu se v každém hradle porovnají všechny množiny vstupních signálů hradla. Při nalezení rozdílů vzniká nová základní smyčka rekonvergenčí. Pro větší přehlednost a porovnávání jsou vstupní množiny hradel rozděleny jednak podle vstupů obvodu, ale také podle úrovní, což umožňuje porovnání množin v lineárním čase postupem od první úrovně do hledané změny.

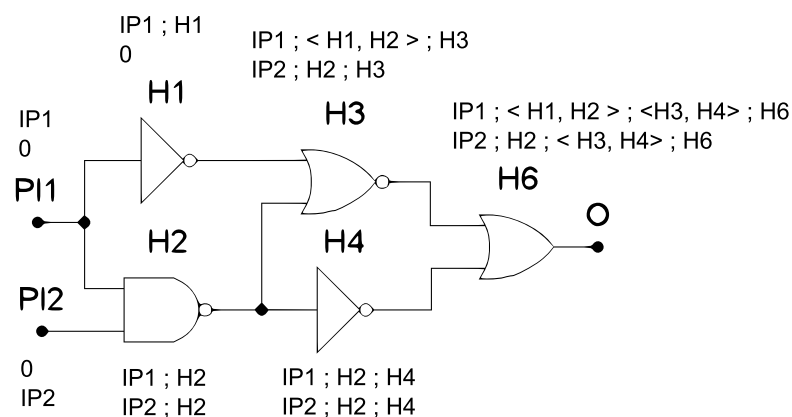
Pro každé hradlo v obvodu se provedou tyto operace:

1. Porovnat vstupní množiny, každý vstup s každým.

- Porovnávají se zvlášť množiny jednotlivých vstupů signálů do obvodu, každá od nejnižší úrovně do konce. Při nenalezení žádného společného hradla dvou množin na jedné úrovni vzniká rekonvergence.
- Počáteční bod se zjistí průnikem hradel o úroveň níže. Velikost průniku určuje počet nových rekonvergenzí.
- Větvení základní smyčky rekonvergence se zjistí projitím obou množin a hledáním hradel, do kterých vede signál z počátečního bodu.

2. Sloučit všechny vstupní množiny.

3. Přidat svoji identifikaci na konec množiny.



Obrázek 3.12: Ukázka výpočtu rekonvergence

Nejlépe je to ukázat na malém příkladu. Na obrázku 3.12 je zmenšený obvod z obrázku 2.4, na kterém se ukazovaly základní smyčky rekonvergence. Z obvodu zmizela hradla H5 a H7, spolu s jednou smyčkou. Nyní se tam nacházejí dvě.

V obvodu jsou dva vstupní signály PI1 a PI2, a proto každé hradlo bude zpracovávat dvě množiny. Počátek celého algoritmu je právě u oněch vstupů. Zde se vytvoří počáteční množiny $\begin{pmatrix} PI1 \\ 0 \end{pmatrix}$

pro PI1 a $\begin{pmatrix} 0 \\ PI2 \end{pmatrix}$ pro PI2. Následující hradlo H1 je invertor, který má pouze jeden vstup. Nedojde k žádnému porovnávání a pouze se vytvoří kopie svého předchůdce a na konec se přidá vlastní identifikace $\begin{pmatrix} PI1; H1 \\ 0 \end{pmatrix}$. Hradlo H2 má sice dva vstupy, ale množiny na nich se neprotínají, proto

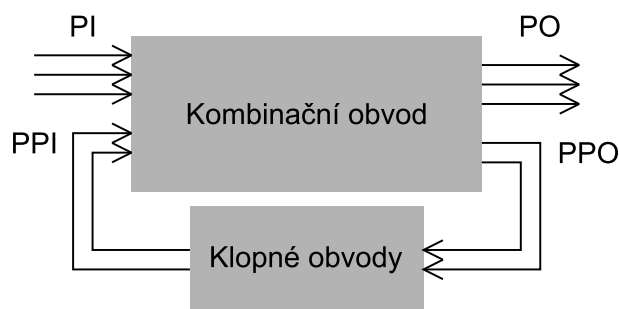
se pouze sloučí a přidá se identifikátor $\begin{pmatrix} PI1; H2 \\ PI2; H2 \end{pmatrix}$. V H3 je nutné porovnávat množiny, první je $\begin{pmatrix} PI1; H1 \\ PI1; H2 \end{pmatrix}$ s $\begin{pmatrix} PI1; H2 \\ PI2; H2 \end{pmatrix}$. K žádnému průniku nedojde při střetu H1 s H2, tudíž počáteční bod smyčky rekonvergence je v tomto případě signál PI1. K získání větvení se obě množiny sloučí a porovnávají se s hradly, kam signál PI1 směřoval, což odpovídá H1 a H2. Konec rekonvergence je současné hradlo H3. K druhému porovnání nedojde, protože jedna skupina je prázdná. Zbývá již jen vytvoření vlastních množin pro H3 $\begin{pmatrix} PI1; \langle H1, H2 \rangle; H3 \\ PI2; H2; H3 \end{pmatrix}$. U invertoru H4 je situace obdobná jako u H1, pouze se

zkopírují množiny a na konec se přidá název aktuálního hradla ($PI1;H2;H4$ / $PI2;H2;H4$). Posledním logickým prvkem v obvodu je H6. Při porovnávání prvních množin ($PI1;(H1,H2);H3$) s ($PI1;H2;H4$) dojde k nulovému průniku až na třetí pozici. Průnik předchozí pozice odhalí počáteční signál H2 a z jeho nástupců se potvrdí větvení signálů H3 a H4. Máme tedy druhou smyčku rekonvergence. Zbývá porovnat druhou množinu z hradel H3 a H4, tedy $PI2;H2;H3$ s $PI2;H2;H4$. Zde dojde k nalezení duplicitní rekonvergence, která se nepřidá.

Algoritmus je celkem výpočetně náročný. K výpočtu v každém logickém členu je zapotřebí hned několika množinových operací.

3.9 Sekvenční obvody

Sekvenční obvod se skládá z kombinační části a klopných obvodů. Kombinační část již umíme vytvořit a proto nebude nic složitého na přidání možnosti generování klopných obvodů. Od nich se požaduje dodržení vstupního parametru, který určí, kolik úrovní kombinační logiky bude mezi vstupem a výstupem i mezi jednotlivými klopnými obvody.



Obrázek 3.13: Princip sekvenčního obvodu

Na obrázku 3.13 je vidět základní schéma sekvenčního obvodu. Kombinační obvod je rozšířen o vstupy a výstupy v počtu přidaných klopných obvodů, což je jednoduchá paměť. Ta umožní uchovat stav signálu do příštího hodinového taktu. Úprava stávajících postupů je malá, postačí zvýšení počtu vstupů a výstupů o vstupní hodnotu počtu klopných obvodů. Ke každému takto přidanému výstupu bude opět náležet hodnota o počtu úrovní kombinační logiky mezi vstupním a výstupním signálem. Opět ale platí, že nejvyšší počet úrovní je omezen velikostí samotné kombinační logiky. Lze například nastavit velikost kombinační části na dvanáct a generovat takový obvod, který bude mít jeden výstup a jeden klopný obvod. Lze nastavit generování tak, aby šířka dvanáct byla pro klopný obvod a samotný výstup může vycházet z druhé úrovně. Systém generování se sám postará o co největší šíření vstupních signálů a tím i signálu z klopného obvodu.

Kapitola 4

Implementace

V minulých kapitolách byly popsány všechny problémy, jejich řešení a byly teoreticky popsány postupy na implementaci. Nyní se zaměříme přímo na popis struktury programu a na podrobnou implementaci složitějších algoritmů.

4.1 Programovací jazyk

Program bude vytvořený v programovacím jazyku C++ [6]. Tento jazyk byl zvolen z několika důvodů. První je fakt, že aplikace nebude obsahovat uživatelské rozhraní, bude se tedy jednat o čistě konzolovou záležitost, kde se hledí hlavně na rychlost zpracování. C++ je objektový jazyk, který je ze všech hlavních jazyků dneška nejbližší k hardwarové reprezentaci, proto se používá na vytváření ovladačů a jiných programů, kde je brána v potaz rychlost. Druhým důvodem je jeho všeobecná podpora v podobě kompilátorů na unixových serverech, kde s tímto programem budu provádět nezbytná měření. Třetím a posledním důvodem je, že mohu použít některé části jádra BOOM [2] pana Ing. Petra Fišera Ph.D., které se zabývá právě touto problematikou.

4.2 Výstupní formát

Jako výstupní formát byl určen Berkeley Logic Interchange Format neboli BLIF [5]. Jedná se o standardní formát zápisu logických obvodů, který se běžně používá. Jeho výhodou je jednoduchost. Jedná se pouze o textovou formu zápisu s jednoduchými parametry. Proto není problém jakkoli programově ztvárněný obvod jednoduše převést do této formy. Další výhodou je jeho podpora sekvenčních obvodů.

Soubor tohoto formátu se skládá z hlavičky, která obsahuje název modelu a také seznam jmen vstupních a výstupních signálů. Následuje definice klopných obvodů pomocí vstupu a výstupu, a také prvotní nastavení.

Tělo souboru je složeno z logických bloků, které mají pouze jeden výstup. Uvnitř bloku je jeden nebo více řádků součinových termů, určujících nulu nebo jedničku výstupového signálu.

To je ve stručnosti základní popis tohoto formátu, který se použije. Podrobnější popis se nachází v příloze B nebo v použité literatuře [5].

4.3 Vstupní formát

Důležitým prvkem programu je určení cesty k získání vstupních parametrů. Obecně lze u konzolové aplikace zvolit tři různé způsoby.

- **Vytvoření zadávacího menu**

V programu se vytvoří textové menu s volbami a jednotlivými možnostmi zadávání. Poté se postupně zadají do konzole jednotlivé údaje.

- **Parametry programu**

Zadání vstupních požadavků pomocí parametru při spuštění. Jedná se o implementačně nejjednodušší a hojně používaný způsob, kdy se do spouštěcího příkazu přidají i parametry.

- **Externí textový soubor**

Zadávání vstupních požadavků pomocí strukturovaného textového dokumentu je pro tento typ programu nejlepší volbou. Soubor se bude dodávat programu pomocí jednoho vstupního parametru a bude mít strukturu jednoduchou pro zápis i následné dekódování.

První možnost je pro uživatele velice zdoluhavá a pro tento typ projektu absolutně nevhodná. Používá se většinou u jednoduchých aplikací, kde se předpokládá malá informovanost uživatele, ne jako v tom to případě specifického problému. Parametry programu jsou lepší volbou, dovolují tvořit skripty pro vytváření různých sérií souborů, ale v našem případě se jedná o velké množství parametrů, jejich pořadí není nijak specifikováno a je tu možnost velké variability zadání. Muselo by se pro přehlednější použití přepínačů, čímž by množství parametrů ještě přibývalo.

V tomto projektu bude využito poslední třetí možnosti. Externí textový soubor bude nejvhodnější, z důvodů přehlednosti a uživatelského pohodlí. Výsledná skladba souboru je popsána v příloze C Uživatelská příručka.

4.4 Hlavní datové struktury

Požadavky na uchování dat jsou celkem malé. První datová struktura, která je zapotřebí, je pro uchování hradel. Je zapotřebí mřížková struktura s přímým přístupem ke každé buňce. Z těchto důvodů jsem zvolil dvojité vložení do sebe `std::vector`, tedy přesně `vector< vector< Hradlo * > >`. Tato struktura je vytvořena a průběžně měněna pouze v jedné metodě `Gener::generovani(Vstup *)`. Pro využití v jiných metodách je posílána jako parametr.

Druhou funkční strukturou je instance třídy `Vstup`, která v sobě uchovává vstupní parametry obvodu. Jedná se o jediný parametr generovací metody a dále je distribuována do vedlejších metod formou parametru. Hlavní třída tedy nemá žádné proměnné.

4.5 Popis tříd

Nyní si projdeme jednotlivé třídy, jejich účel a nejdůležitější metody, abychom se později mohli blíže podívat na hlavní algoritmy programu.

4.5.1 Class Hradlo

Nejdůležitější je třída `Hradlo`, která uchovává všechny potřebné informace (typ, vstupy, výstupy, úroveň atd.) a má tři různé funkce. První je klasické hradlo, druhá je reprezentace vstupu a poslední je zástupný výstup. Rozdíl mezi těmito typy je dán pozicí v hlavní mřížce. Na první úrovni jsou pouze vstupy a na poslední jsou pouze výstupy nebo zástupná hradla, identifikovatelná pomocí úrovně nastavení úrovně na -1 .

Důležité metody této třídy:

- `void vypoctiPozorovatelnostCOP()`
Výpočet pravděpodobnosti zda signál je pozorovatelný na výstupech.
- `void vypoctiRiditelnostCOP()`
Výpočet pravděpodobnosti, zda signál má hodnotu 1.
- `void vypoctiRekonvergence()`
Vypočte z informací svých předchůdců rekonvergence.

4.5.2 Class Gener

Hlavní třída programu zapouzdřuje všechny metody spojené s hledáním požadovaného obvodu. Jako jediná neobsahuje žádné proměnné, pouze metody. Jediným úkolem třídy je vygenerovat obvod nebo získat informace z externího souboru. Stav udržuje metoda `generovani(Vstup *)`, ve které je obsaženo několik hlavních algoritmů a z ní se volají ostatní metody.

- `void generovani(Vstup * param);`
Hlavní metoda, která postupně vytváří hledaný obvod, a volá ostatní metody. Zde se vytvoří mřížka hradel, vytvoří se jednovýstupový obvod a dovytvoří se ostatní výstupy.
- `int nastaveniSirkyObvodu(Vstup * param);`
Metoda na určení šířky mřížky ze vstupních parametrů.
- `void pojmenovaniObvodu(int pocet, int lev, v< v< Hradlo*> > * uobv);`
Projde a unikátně pojmenuje veškerá hradla, která patří do obvodu.
- `void getRekonvergence(vector< vector< Hradlo*> > * obvN, Vstup * param);`
Metoda starající se o výpočet rekonvergenzí, které se vrátí v datové struktuře uvnitř instance třídy `Vstup`.
- `void snizVetveni(vector< vector< Hradlo*> > * obvN, Vstup * param, ...);`
Podle vstupních parametrů se pokusí o snížení maximálního a průměrného větvení.

- `void ulozOBV(vector< vector< Hradlo* > > * obv, Vstup * param);`
Uloží obvod do textového formátu blif.
- `Vstup blifinfo(Blif * b);`
Z načteného souboru zjistí všechny parametry a vypočítá rekonvergenční.

4.5.3 Class GenerVstupu

Nyní se dostáváme k vedlejším třídám, první z nich je `GenerVstupu`, která jediný účel, generovat vstupní signály pro jednotlivá hradla. Objekt této třídy se vytvoří pro každou úroveň obvodu a poté dodává signály všem hradlům na úrovni. Metody jsou následující:

- `GenerVstupu(vector< vector< Hradlo* > > * obv, int nUroven, int navaz);`
Konstruktor, ve kterém se připraví množiny vstupů platící pro celou úroveň.
- `vector<pair<int, int> > * getSignal(int pocet, bool prijato);`
Metoda, jež vrátí požadovaný počet vstupů, nalezených podle pravidel. Hodnota typu boolean slouží k informaci, zda bylo předešlé hradlo přijato nebo vyhozeno z důvodů udržení aktuálního pokrytí vstupními signály.

4.5.4 Class Vstup

Malá třída, uchovávající parametry generovaného obvodu. Je tvořena konstruktorem, který slouží ke dekódování vstupního souboru, a z jednotlivých parametrů, které jsou přímo přístupné.

4.5.5 Class RekoObj

Třída na uchování nalezených rekonvergenzí v obvodu. Má pouze dvě metody.

- `RekonObj(Hradlo * signal, set <Hradlo * > vetveniSet , Hradlo * ukonceni);`
Konstruktor třídy, slouží ke uložení informací o právě nalezené rekonvergenzi.
- `int rsize();`
Metoda vracející informaci o velikosti rekonvergenční.

4.5.6 Class RekonBunkaSeznam

Třída, uchovávající jednotlivé informace o větvení signálů. Každé hradlo má jednu instanci této třídy. Slouží také k slučování větvení z předcházejících hradel a vytváření rekonvergenzí.

- `void slouci(RekonBunkaSeznam * rekbunSeznam);`
Přičte k současné množině novou množinu.
- `void porovnej(RekonBunkaSeznam * rekbunSeznamA, ...);`
Metoda na postupné porovnání množin hradel všech vstupních signálů obvodu.
- `void pridatHradlo(Hradlo * pridat);`
Na konec všech množin přidá aktuální hradlo.

4.5.7 Class RekBunka

Obsahuje reprezentaci jedné samotné množiny hradel navazujících na jeden vstup. V této třídě dochází ke identifikaci smyček rekonvergence.

- `void sloucit(RekBunka * nova);`
Sloučí novou buňku s aktuální.
- `void porovnej(RekBunka * nova, Hradlo * aktual, list<RekonObj*> * rek);`
Porovná dvojici množin hradel a vyhledá rekonvergence, které přidá do celkové množiny.
- `void pridejNaKonec(Hradlo * aktual);`
Přidá na konec množiny nové hradlo.

4.6 Náhodný generátor vstupů hradla

Ač se to může zdát jako jednodušší algoritmus, není generování vstupů pro jednotlivá hradla tak jednoduché. Jsou zde kladeny velké požadavky na výběr vstupu, které byly již popsány podrobně kapitole 3.3, ale také na rychlost zpracování, aby zbytečně nedocházelo ke zpomalení generování hlavní mřížky.

Za tímto účelem byla vytvořena samostatná třída. Ta umožňuje vytvoření instance pro každou generovanou úroveň mřížky, čímž se nemusí vytvářet množiny množných vstupů při každém požadavku na vstupy hradla.

Nejdříve si popíšeme konstruktor. Jako parametr se zde vyskytuje již vytvořená mřížka hradel, číslo aktuálně tvořené úrovně a taktéž návaznost vstupních signálů. Z těchto parametrů se vytvoří tři množiny možných souřadnic typu `vector<pair<int, int> >`. První je množina pro první vstup hradla `mnoz1`, ta se skládá ze souřadnic všech hradel na předešlé úrovni. Druhá `mnoz1bezINV` je také pro první vstup, ale pouze s tou změnou, že neobsahuje žádné souřadnice invertorů na minulé úrovni. Poslední třetí množina `mnoz2` obsahuje všechny souřadnice dosažitelných hradel, podle parametru návaznosti.

```
GenerVstupu( vector< vector< Hradlo* > > * obv,int novaUroven, int navaznost){
1   Inicializuj množiny
2   Projdi úroveň mřížky od (nové úrovně -1) do nuly {
3       Projdi jednotlivě úroveň {
4           Vlož souřadnici do mnoz2
5           Když je souřadnice z minulé úrovně {
6               Vlož do mnoz1
7               Když není hradlo INV {
8                   Vlož do mnoz1bezINV
9               }
10          }
11      }
12  }
}
```

Konstruktor pouze projde hradla v dosahu návaznosti a vloží je do množin pro vyhledávání vstupů, asymptotická složitost je závislá na šířce mřížky S a návaznosti signálů N .

$$O(S \cdot N) \quad (4.1)$$

Počet prvků nutných k projití není vždy $S \cdot N$. Při vytváření menší úrovně, než je hodnota návaznosti, není maximálního počtu dosaženo.

Získávání vstupních signálů pro vstup hradel probíhá přes metodu `getSignal`, jejichž parametry jsou počet požadovaných signálů a boolovská proměnná, zda bylo minulé hradlo začleněno do mřížky. Generování množiny vstupních signálů probíhá podle pravidel, které byly popsány na straně 11.

```
vector<pair<int,int> > * getSignal(int pocet,bool minulehradloprijato){
1   Když nebylo minule hradlo přijato
2       Vrat' pokryti úrovně a hradla no minulé hodnoty
3   Když je počet vstupů == 1 {
4       Vybírej vstup z množiny mnoz1bezINV
5   }jinak{
6       Vstup z množiny mnoz1
7   }
8   Když pokrytí úrovně != 1 {
9       Projdi náhodně množinu pro první vstup
10      Když rozšíří pokrytí přijmi ho
11  }jinak{
12      Náhodně vyber první vstup
13  }
14  Pro zbylé vstupy postupuj následovně{
15      Když pokrytí úrovně není kompletní
16          Projdi náhodně mnoz2
17              Když rozšíří, přijmi ho
18      Když pokrytí hradla není kompletní
19          Projdi náhodně mnoz2
20              Když rozšíří a není to první vstup, přijmi ho
21      Pokud stále nebyl nalezen vstup, vyber náhodný z mnoz2
22  }
}
```

První vstup se vytváří z množiny `mnoz1`, pokud je ale požadavek pouze na jeden signál, jedná se o generování invertoru. Z tohoto důvodu se použije `mnoz1bezINV`. První vstup se pokusí rozšířit pokrytí úrovně. Pro získání ostatních vstupů je zde `mnoz2`. I druhý vstup se pokusí zvýšit pokrytí úrovně, ale pokud je již kompletní nebo takový vstup neexistuje, přejde se k hledání signálu pro zvýšení pokrytí hradla a následně k náhodnému výběru. Podmínky generování popsané na straně 11, jsou ve skutečnosti moc přísné. Při hledání jednoho vstupu nenastane nikdy situace, při které by se musela prohledat více jak jedna množina a ve většině případů se pouze náhodně vyberou signály.

Pokud pokrytí hradla či úrovně není úplné, existuje vždy minimálně jeden signál, který ho rozšíří. Tudíž se signál nalezne vždy v první množině. Hradla na nižších úrovních rozšíří svá pokrytí co

nejrychleji na maximum, při kterém následující úrovně již neprohledávají žádné množiny, pouze náhodně zvolí signál. Asymptotická složitost proto není $O(n \cdot |mnoz|)$, kde n je počet požadovaných vstupů, ale v naprosté většině případů pouze lineární na počtu vstupů hradla.

$$\theta(n) \tag{4.2}$$

4.7 Generování mřížky

Nejdůležitějším algoritmem v celém programu je generování mřížky hradel. Na počátku se inicializuje mřížka složená z vektorů `stl::vector`, poté se do ní vloží na nultou úroveň všechny potřebné vstupní signály, jak klasické vstupy, tak i případně výstupní signály klopných obvodů. Následuje získání šířky mřížky metodou `Gener::nastaveniSirkyObvodu`. Generování probíhá po jednotlivých úrovních, ve které se vždy vytvoří nová aktuální instance třídy `GenerVstupu`, jež bude dodávat vstupní signály pro hradla.

Samotné hradlo vzniká počátečním výběrem typu, kde je zohledněna i potřebná četnost. Ta je zajištěna náhodným výběrem z vektoru obsahující takový počet jednotlivých typů, jako je hodnota ve vstupním parametru četnosti.

Následuje náhodné určení počtu vstupů hradla v závislosti na jeho vybraném typu. Generátor vstupů dodá vstupní signály, které se na hradlo naváží. Výpočtem říditelnosti získáme informaci, zda hradlo splňuje kritéria. Pokud ano, hradlo je přijato a vloží se do mřížky, pokud ne, dojde k zahodění hradla a novému generování. Je zde také ochrana proti zacyklení v případě, že by docházelo k zahazování hradel způsobené nízkou hodnotou říditelnosti.

Tvorba mřížky je zastavena jednu úroveň před konečným počtem a následuje hledání přesného počtu hradel.

Pseudokód generování mřížky:

```

1   Inicializuj mřížku
2   Vlož přesný počet vstupů do mřížky na úroveň 0
3   Získej šířku mřížky
4   Vytvářej jednotlivé úrovně od 1 do (počtu úrovní - 1){
5       Vytvoř instanci generátoru vstupních signálů
6       Vytvářej hradla na aktuální úrovni v počtu od 0 do šířky mřížky{
7           Podle četnosti náhodně zvol typ hradla
8           Podle typu a max. počtu vstupů náhodně zvol počet vstupů
9           Požádej generátor vstupů o signály
10          Vypočítej říditelnost
11          Pokud říditelnost projde{
12              Vlož hradlo do mřížky
13          }jinak{
14              Zahod' hradlo a generuj znovu
15          }
16      }
17  }
```

Asymptotická složitost generování mřížky je dána šířkou S a počtem úrovní L .

$$\theta(L \cdot S) \quad (4.3)$$

4.8 Hledání přesného počtu hradel

Hledání přesného počtu logických prvků navazuje na algoritmus generování mřížky v hlavní metodě `Gener::generovani`. Jedná se o procházení všech kombinací vstupních signálů pro konečné hradlo na požadované úrovni. Jak bylo popsáno v kapitole 3.4 str. 14, počet dvojic signálů je maximálně $p = S \cdot S \cdot (N - 1) + \binom{S}{2}$. Při prohledávání takto velkého stavového prostoru se postupuje od signálů na vyšších úrovních k nižším, tím se nejdříve prohledávají možnosti s teoreticky vyšším počtem hradel a postupuje se k nižším počtům. Algoritmus počítání je průchod mřížky do hloubky z vybraných signálů, který má složitost závislou na počtu hradel n tedy $\theta(n)$.

Při nalezení požadovaného množství se vytvoří dvouvstupové hradlo a vypočítá se jeho řiditelnost. Pokud hradlo splňuje podmínky řiditelnosti a kompletního pokrytí vstupních signálů, je přijato. V případě nedodržení míry řiditelnosti se projdou ostatní typy, zda některý míru splňuje. Pokud po projití všech kombinací nedojde k nalezení požadovaného množství, obvod nebyl nalezen a program se ukončí.

Pseudokód hledání přesného počtu hradel:

```

1   Projdi všechny úrovně X od poslední do nejnižší možné{
2       Projdi všechny kombinace dvojic signálů úrovně X s poslední úrovní{
3           Spočítej počet hradel
4           Když je počet správný
5               Vytvoř dvouvstupové hradlo
6               Když je pokrytí vstupů je jedna a projde řiditelnost{
7                   Přijmi hradlo a vlož ho do mřížky
8               }jinak{
9                   Vyzkoušej změnu typu, zda projde řiditelnost
10                  Pokud ne, hledej dál
11              }
12          }
13      }
14  }
```

Výpočetní náročnost je dána počtem kombinací, které je nutno projít. Řádově se jedná o maximálně $S^2 \cdot N$ možností. Pro každou možnost je nutné zjistit počet nalezených hradel h . Celková složitost je:

$$O(S^2 \cdot N \cdot h) \quad (4.4)$$

4.9 Dovy tvoření výstupů

Poslední z algoritmů, obsažených v hlavní generující metodě je tvorba dodatečných výstupů obvodů. Ta využívá dvou funkcí. První je na zjištění počtu hradel a jejich konkrétních úrovní potřebných k vytvoření jednoho výstupu. Jednoduchý výpočet používá celou množinu výstupů, konkrétně jejich cílových úrovní. Pokud je právě generovaný výstup ve stejné úrovni jako předešlý, je použito k vytvoření o jedno hradlo více. V opačném případě je použito pouze jedno hradlo. Úrovně potřebných hradel jsou v rozmezí maximální návaznosti signálu.

Druhou použitou funkcí je získání množiny hradel z původního jednovýstupového obvodu, splňující rozsah možných úrovní pro navázání vstupu. Jedná se o prohledávání jednovýstupového obvodu do hloubky s náročností $O(n)$, kde n je počet hradel v obvodu.

Algoritmus pro každý jednotlivý výstup získá množinu úrovní, na které se vloží dvouvstupová hradla náhodného typu. První vstup je signál vždy z úrovně o jedna menší než právě vkládané a pochází z původního obvodu. V případě vkládání prvního hradla z množiny je druhý signál opět z původního obvodu, avšak rozsah úrovní je dán maximální návazností. Pokud je hradlo v množině nejméně druhé, druhý vstup pochází z jeho předchůdce. Pro vložení hradla do mřížky se musí splnit kritéria říditelnosti, pokud tomu tak není, projde se množina dvouvstupových typů hradel. V případě, že se nenalezl žádný typ splňující podmínky, generování hradla začne znovu. Po vytvoření celé množiny hradel k jednomu výstupu se musí samotný výstupní signál vyvést pomocí zástupného hradla do poslední úrovně mřížky, pokud již daný výstup této úrovně nedosahuje.

Pseudokód vytváření výstupů:

```

1   Vytvoř zbylé výstupy{
2       Pro každý výstup vypočítej nutný počet hradel a jejich úrovně
3       Vytvoř množinu hradel na definovaných úrovních
4       Projdi množinu{
5           První vstup je vždy z úrovně o jedna menší
6           Když je hradlo první z množiny pro vytvoření vstupu{
7               Druhý vstupní signál z libovolné úrovně omezené návazností
8           }jinak{
9               Navaž na předešlé vložené hradlo
10          }
11          Náhodně zvol typ hradla
12          Vypočítej říditelnost
13          Když je říditelnost v požadovaných mezích{
14              Vlož hradlo do mřížky
15          }jinak{
16              Vyzkoušej změnu typu, zda projde říditelnost
17              Pokud ne, hledej dál
18          }
19      }
20 }
```

Složitost tohoto algoritmu není výrazná z důvodu relativně malého počtu výstupů. Znak H udává počet hradel v jednovýstupovém obvodu a Hn je počet nových hradel hradel vytvářející výstupy.

$$O(H \cdot Hn) \quad (4.5)$$

4.10 Úprava větvení signálů

Algoritmus pro úpravu větvení signálu se skládá ze dvou samostatných celků podle dvou různých parametrů. První nastavení se týká větvení signálu. Tato hodnota udává počet hradel, do kterých jednotlivý signál vchází. Nastavení tohoto parametru, aby byla splněna podmínka maximálního povoleného větvení signálu obvodu, se provede následovně:

Nejdříve se projde obvod a jednotlivé signály se vloží do množiny sestupně podle počtu větvení. Následuje hlavní cyklus algoritmu, který se ukončí ve chvíli, kdy je splněna podmínka maximálního větvení nebo pokud již nelze větvení snížit. V těle cyklu se vždy provedou následující kroky:

Vybere se signál z množiny s největším větvením a náhodně se prochází všechny jeho větve, pro které se algoritmus pokusí najít náhradní signál, jenž by je v následujícím hradle nahradil. K tomu se použije již vytvořená množina signálů, kterou se prochází od signálů s nejnižším větvením až po signály, jenž mají o dva menší počet než právě snižovaný signál. Pro přijetí změny větvení musí zástupný signál splňovat podmínku maximální délky signálu, nebo-li návaznost. Druhá podmínka zajistí zachování úrovně hradla, nesmí se změnit signál z předešlé úrovně pokud zbývá pouze jeden takový.

Po záměně signálu se provede v celém obvodu přepočítání říditelnosti a pokrytí hradel jednotlivými vstupními signály. Pro konečné přijetí se musí říditelnost výstupních signálů obvodů pohybovat v daných mezích a všechny výstupní signály musí být společně závislé na všech vstupních, pokud ne změna se vrátí a dál se pokračuje v hledání. Pokud po projití všech větví signálu nedojde ke snížení, vyzkouší se možnost odstranit jeden vstup hradla. To je však možné pouze za předpokladu, že signál ústí do hradla s třemi a více vstupy. Pokud se ani nyní větvení nezmění, označí se signál značkou, která ho odstraní z přidání do množiny. Na konci cyklu se provede opětovně vytvoření množiny podle počtu větvení bez označených signálů.

Pseudokód snižování maximálního větvení signálů

```

1   Vytvoř seřazenou množinu podle počtu větvení
2   Opakuj, dokud není maximální počet větvení v toleranci{
3       Vyber signál s největším větvením
4       Postupně projdi všechny větvení signálu
5           Projdi všechny signály množiny, které
6               mají nižší větvení než max. možné.{
7               Když signál splňuje podmínky{
8                   Nahrad' signál novým
9                   Přepočítej říditelnost a pokrytí výstupů
10                  Pokud je říditelnost v pořádku a
11                      pokrytí výstupů je úplné{
12                          Přijmi změnu
13                          Ukonči snižování
14                      }jinak{
15                          Vrat' změnu

```

```

16         }
17     }
18 }
19     Pokud nedošlo ke snížení větvení signálů
20         Zkus jedno větvení odstranit snížením počtu vstupů hradla
21         Pokud to nepomůže označ signál, aby se nezapočítal
22     }
23     Vytvoř novou seřazenou množinu větvení
24 }

```

Asymptotická složitost tohoto algoritmu je dána z velké míry počtem signálů, překračující maximální povolené větvení Xh a z počtu jejich následovníků oh . Každý takový signál je buď snížen a je možné, že jeho hodnoty stále překračují danou mez, tudíž se bude procházet znovu nebo nenastane změna a signál z množiny zmizí. Pro každý výstup se musí v nejhorším případě projít všechny signály s nižším větvením Nh . Při nalezení vhodného signálu se musí přepočítat říditelnost a pokrytí, je tedy nutné ještě dvakrát projít celou množinu hradel v obvodu h . Na konci zpracování hradla se vždy vytvoří množina nová.

$$o(Xh \cdot oh \cdot Nh \cdot h + Xh \cdot h) \doteq O(h^2) \quad (4.6)$$

Při střídmém nastavení tohoto parametru není množina Xh nikterak velká a počet větvení oh je možné zanedbat. Za těchto předpokladů vychází kvadratická složitost v závislosti na počtu hradel.

Druhým algoritmem, zabývajícím se větvením, je snižování průměrného větvení signálu v obvodu. Jak již bylo řečeno v kapitole 3.11 str. 25, není možné snížit tuto hodnotu při použití maximálně dvouvstupových hradel. Za předpokladu tří a více vstupů dojde ke snížení průměrného větvení signálu odstraňováním vstupů do logického hradla, minimálně však na hodnotu dva.

Algoritmus nejdříve vytvoří množinu všech hradel se třemi a více vstupy. Následuje cyklus, který končí buď vyprázdněním množiny nebo splněním požadavku na tuto hodnotu. Náhodně se vybírají z množiny hradla, kterým se následně projdou všechny jeho vstupy. Vstup vhodný pro odstranění musí splňovat podmínku původu signálu s větším větvením než jedna. Tímto se zabrání odříznutí hradla od vazby na výstupní signály. Druhou podmínkou je existence minimálně jednoho jiného vstupu do hradla z minulé úrovně.

Po odstranění vstupu je nutné znovu přepočítat říditelnost a pokrytí vstupními signály ve všech logických prvcích obvodu. Výstupní signály se i nadále musí pohybovat ve stanovených mezích říditelnosti, jinak je nutné signál vrátit.

Pseudokód snižování průměrného větvení signálů

```

1   Projdi obvod a vytvoř množinu s hradly o počtech vstup > 2
2   Opakuj dokud není množina prázdná nebo
3       dokud není větvení v požadovaném rozsahu{
4       Vyber náhodné hradlo
5       Najdi takový vstup, který není jediný z minulé úrovně
6       a který má více jak jedno větvení

```

```

7         Odstraň ho
8         Pokud projde omezení říditelnosti{
9             Odstraň hradlo z množiny
10        }jinak{
11            Vrat' nazpět signály
12    }
```

Náročnost tohoto algoritmu je menší než snižování maximálního větvení. Množina hradel se vytváří pouze jednou, poté se prochází jejich vstupy, kterých je relativně málo a je možné je zanedbat. Řádově vychází opět kvadratická závislost na počtu hradel h .

$$O(h^2) \quad (4.7)$$

Předchozí dva algoritmy se nacházejí v metodě `snizVetveni` za sebou. Metoda je volána z hlavní metody `Gener::generovani` po kompletním vytvoření obvodu, za předpokladu existence požadavku na úpravu těchto parametrů. Metoda mění strukturu obvodu, tudíž dochází k několikanásobnému spuštění metody dokud nedojde ke splnění podmínek nebo již nedochází ke změně parametrů.

4.11 Výpočet smyček rekonvergenčí

Výpočet počtu smyček rekonvergenčí probíhá na čtyřech úrovních. Na nejvyšší úrovni, reprezentovanou funkcí `Gener::getRekonvergence`, se postupně zavolá výpočet u všech hradel v obvodu. Funkce `Hradlo::vypoctiRekonvergeneci` je volána od hradel nejnižší úrovně a postupuje se až k nejvyšší úrovni.

Jednotlivé hradlo disponuje vlastním seznamem rekonvergenčí `list<*RekonObj>` a také seznamem s jednotlivými množinami posloupností hradel pro výpočet `RekonBunkaSeznam`.

Hradlo obsahuje vždy právě tolik posloupností, kolik je vstupních signálů. Tato posloupnost je reprezentována třídou `RekonBunka` a představuje cestu vstupního signálu k aktuálnímu hradlu. Jsou zde všechna hradla, kudy signál prošel, rozříděná podle svých úrovní.

Algoritmus výpočtu rekonvergenčí na úrovni hradla probíhá následovně. Nejdříve se zjistí, zda se nejedná o hradlo zastupující vstupní signál. Pokud ano, vytvoří se počátek posloupnosti signálu. Následně se funkce ukončí. U zbylého počtu hradel dojde k inicializaci seznamu vytvořených rekonvergenčí. Poté se provede porovnání seznamu posloupností všech vstupů každý s každým. Při tomto porovnání dochází k vytvoření rekonvergenčí. Zbývá vytvořit vlastní seznam sloučením všech vstupních seznamu posloupností. A následně doplnit na konec posloupností svůj odkaz.

```

Peudokód Hradlo::vypoctiRekonvergeneci(...) {
1     Když se jedná o vstupní signál obvodu{
2         Vytvoř počáteční množinu
3         Konec
4     }
5     Inicializuj množinu nalezených rekonvergenčí
6     Porovnej instance objektu RekonBunkaSeznam každého vstupu hradla s každým
7     Sluč objekty vstupů dohromady.
}
```


Na úrovni instance `RekonBunkaSeznam` reprezentující množinu posloupností `RekonBunka` se provede pouze replikace příkazů *sluč* a *porovnej* z vyšší úrovně na všechny posloupnosti.

Nejnižší úroveň je samotná posloupnost tzv. buňka, na které se provádí samotné hledání smyček rekonvergenčí. Hlavní metodou je zde `RekonBunka::porovnej`, kde dojde k porovnání dvou posloupností. Pokud je jedna z buněk prázdná, ještě neexistuje vazba mezi tímto hradlem a vstupním signálem, porovnání se ukončí. Jinak následuje postupné procházení buněk od nejnižší úrovně po nejvyšší. Pokud nedojde k žádnému průniku na aktuální pozici a zároveň existuje průnik na předchozí pozici, dochází k nalezení rekonvergenčí. Průnik reprezentuje množinu signálů, kde vzniká smyčka. Pro každý signál se projdou obě posloupnosti a identifikují se větve signálu tvořící rekonvergenčí. Všechny parametry rekonvergenčí jsou zjištěny, ale je nutné zkontrolovat, zda již tato rekonvergenčí nebyla vložena projitím seznamu vzniklých rekonvergenčí.

```
Pseudokód RekBunka::porovnej(...) {
1   Pokud je jedna z buněk prázdná{
2       Konec
3   }
4   Procházej buňku
5       Jestliže neexistuje průnik dvou porovnávaných buněk v pozici X{
6           Vytvoř průnik pozice X-1
7           Projdi všechny prvky průniku{
8               Najdi větve tvořící rekonvergenčí
9               Projdi naposled vytvořené rek.
10              Když tato rek. neexistuje přidej
11          }
12      }
}
```

Sloučení dvou buněk posloupností probíhá postupným procházením a vkládání jednoho okénka do druhého. Jeden prvek reprezentující prvky na stejné úrovni je tvořen množinou `stl::set`, která má logaritmickou složitost vkládání, ale zamezí vytváření duplicit.

Celková složitost tohoto algoritmu se komplikovaně určuje. Při výpočtu u jednoho hradla se musí porovnat každý vstup s každým, tedy $\binom{H_{pi}}{2} = \frac{H_{pi} \cdot (H_{pi} - 1)}{2}$ krát, kde H_{pi} je počet vstupů hradla. Porovnání probíhá na posloupnostech vždy v počtu vstupů obvodu. U jednotlivých posloupností má porovnání lineární závislost na počtu úrovní. Závislost porovnání $PI \cdot L$ je počet vstupu PI krát počet úrovní obvodu L . Množinové operace jsou v každé operaci provedeny na počtu hradel na jednotlivé úrovni Hl .

Počet hradel na úrovni vynásobený počtem úrovní je celkový počet hradel. $L \cdot Lh = H$. Celková složitost vychází:

$$O\left(H \cdot \frac{H_{pi} \cdot (H_{pi} - 1)}{2} \cdot PI \cdot L \cdot Hl\right) = O\left(H^2 \cdot \frac{H_{pi} \cdot (H_{pi} - 1)}{2} \cdot PI\right) \quad (4.8)$$

4.12 Získávání informací

Celý program je vytvořený a nyní již zbývá poslední věc. Algoritmus pro získávání parametrů z jiných souborů typu Blif. Ten se využije pro získání parametrů reálných obvodů a následného vytváření klonů.

Jak již bylo zmíněno, formát se skládá z jednotlivých logických bloků definovanými součinnými termy, které mohou být libovolné. Neexistuje zde konečné množství standardizovaných prvků, proto se nemůže obyčejný soubor použít pro načítání. Nejdříve je nutné použít přemapování na používanou množinu hradel. Využijí nástroje pro logickou syntézu ABC [4], který dokáže přemapovat soubor na logické prvky definované externí knihovnou. Takto upravený soubor se již načte pomocí použitého jádra BOOM a výsledný soubor se bude dále zpracovávat.

Knihovna hradel je ve formátu Cadence genlib. Má jednoduchou textovou strukturu, kterou není problém upravit do potřebné podoby. Každý logický prvek definován názvem, počtem tranzistorů, logickou funkcí a zpožděními vstupních signálů. Počet tranzistorů je zde pro minimalizaci na co nejmenší plochu, což pro tento projekt není důležité, přesto je nutné tento údaj vyplnit. Obdobně je to se zpožděními signálů.

Vytvoříme si vlastní knihovnu, která obsahuje pouze prvky používané tímto programem. Aby bylo možné s touto knihovnou pracovat, je nutné doplnit logické prvky o buffer a konstanty nula a jedna. Bez těchto prvků není možné knihovnu použít.

```
GATE inv1 1 0=!a; PIN * INV 1 999 0.9 0.3 0.9 0.3
GATE nand2 2 0=!(a*b); PIN * INV 1 999 1.0 0.2 1.0 0.2
GATE nor2 2 0=!(a+b); PIN * INV 1 999 1.4 0.5 1.4 0.5
GATE and2 3 0=a*b; PIN * NONINV 1 999 1.9 0.3 1.9 0.3
GATE or2 3 0=a+b; PIN * NONINV 1 999 2.4 0.3 2.4 0.3
GATE xor2 5 0=a!*b+!a*b; PIN * UNKNOWN 2 999 1.9 0.5 1.9 0.5
GATE nxor2 5 0=!(a!*b+!a*b); PIN * UNKNOWN 2 999 2.1 0.5 2.1 0.5
GATE buf1 1 0=a; PIN a NONINV 1 999 1.00 0.00 1.00 0.00
GATE zero0 0 0=CONST0;
GATE one0 0 0=CONST1;
```

Takto vypadá knihovna hradel 2g_all.genlib, ta obsahuje všechna hradla, která při generování prvků s maximálně dvěma vstupy vznikají. Je nutné se rozhodnout, na jaký počet se má logický obvod přemapovat, aby nedocházelo k zbytečně velkému nárůstu hradel z důvodu nedostatku vstupů. Proto je vhodné používat knihovnu 5g_all.genlib obohacenou o hradla až s pěti vstupy.

```
GATE nand3 3 0=a*b*c; PIN * INV 1 999 1.9 0.3 1.9 0.3
GATE nand4 4 0=a*b*c*d; PIN * INV 1 999 1.9 0.3 1.9 0.3
GATE nand5 5 0=a*b*c*d*e; PIN * INV 1 999 1.9 0.3 1.9 0.3
GATE nor3 3 0=!(a+b+c); PIN * INV 1 999 1.4 0.5 1.4 0.5
GATE nor4 4 0=!(a+b+c+d); PIN * INV 1 999 1.4 0.5 1.4 0.5
GATE nor5 5 0=!(a+b+c+d+e); PIN * INV 1 999 1.4 0.5 1.4 0.5
GATE and3 4 0=a*b*c; PIN * NONINV 1 999 1.9 0.3 1.9 0.3
GATE and4 5 0=a*b*c*d; PIN * NONINV 1 999 1.9 0.3 1.9 0.3
```

```
GATE and5 6 0=a*b*c*d*e; PIN * NONINV 1 999 1.9 0.3 1.9 0.3
GATE or3 4 0=a+b+c; PIN * NONINV 1 999 2.4 0.3 2.4 0.3
GATE or4 5 0=a+b+c+d; PIN * NONINV 1 999 2.4 0.3 2.4 0.3
GATE or5 6 0=a+b+c+d+e; PIN * NONINV 1 999 2.4 0.3 2.4 0.3
```

Přesto se knihovny s menším počtem vstupů hradel využijí při minimalizaci vygenerovaných obvodů pomocí programu ABC a následném uložení. Tam se musí použít stejná množina hradel, jako při vytváření obvodu.

Přemapovaný obvod se načte pomocí třídy `Blif` z použitého jádra BOOM. Signály jsou ve třídě `Blif` specifikovány jako v samotném formátu pomocí textových názvů, neexistují zde žádní ukazatelé, kteří by definovali konkrétní vazby. Tato třída se také postará o určení typu, tím je následná transformace na mřížku hradel celkem jednoduchá. Postup je následovný.

Nejdříve se vytvoří hradla definující vstupní signály. Poté se postupně projde množina kombinačních prvků třídy `Blif` a vytvoří se jednotlivá hradla, která jsou uložena do připravené mapy `stl::map<string, Hradlo * >`, pro rychlé hledání prvků podle názvů. Při této příležitosti se spočítají hradla jednotlivých typů. Při tvoření hradel se objeví i prvky, které nelze vytvořit v našem modelu, ale musí se s nimi pracovat.

- **Buffer**

Tento prvek nijak nezasahuje do logické funkce obvodu, proto se může vypustit. Vstupní a výstupní signál bufferu se liší, musí se tedy uložit do mapy nová položka s názvem, ale s již vytvořeným hradlem. Dva signály nyní ukazují na stejné hradlo.

- **Const**

Na konstantní signály lze pohlížet jako na vstupy do obvodu, které se ovšem nesmí započítat do výsledného počtu vstupů.

- **Latch**

Klopné obvody se použijí jako vstupní signály do obvodu a vstup latch jako výstup.

Následně se opět projde množina a naváží se vstupní a výstupní vazby na hradla.

Veškerá hradla máme již vytvořena a navázána, ještě nelze vytvářet mřížku. Neznáme úroveň jednotlivých hradel, pouze vstupních signálů. Projde se tedy množina prvků a postupně se propaguje hodnota úrovně pomocí ukazatelů.

Vytvoří se mřížka a vloží se hradla na správnou úroveň. Poslední úpravou je vytažení výstupních signálů a vstupů latch, pomocí zástupných hradel do poslední úrovně. Nyní máme načtený obvod připraven pro použití informačních metod:

```
int Gener::getNavaznost(vector< vector< Hradlo*> > * obvN );
int Gener::getMaxVstupHradla(vector< vector< Hradlo*> > * obvN);
double Gener::getPrumerVetveni( vector< vector< Hradlo*> > * obvN );
int Gener::getMaxVetveni( vector< vector< Hradlo*> > * obvN );
```

Poslední parametry, které nebyly zjištěny, je počet úrovní mezi vstupy a jednotlivými výstupy a také obdobný parametru klopných obvodů. To je již triviální záležitost, pouze se přečte úroveň výstupních hradel.

Kompletní zjištěné údaje se uloží ve formátu pro opětovné načtení a vygenerování obvodů stejných parametrů.

Spouštění tohoto algoritmu je popsáno v příloze C Uživatelská příručka. Vstupem je již zmíněný přemapovaný obvod a výstupem programu je textový soubor s nalezenými parametry ve formátu pro generování obvodu.

4.13 Porovnání s existujícím řešením

Zadání tohoto projektu je podobné jako u diplomové práce pana Měchury [3], který vytvářel náhodné obvody definované booleovskou funkcí, stavovým automatem nebo sítí hradel. Jeho práce byla zaměřena na dodržení zadaných parametrů testovatelnosti. Při tvorbě sítí hradel, která byla parametrově podobná, využíval genetických algoritmů pro úpravu parametrů. Tato metoda však vedla k velkému nárůstu lehce testovaných hradel typu XOR, NXOR a INV, což nepříznivě přispělo ke snadné minimalizaci obvodů.

Já využívám testovatelnost pouze pro zamezení vzniku tautologie nebo kontradikce a zaměřuji se na tvorbu víceúrovňových sítí hradel s co nejmenší možností minimalizace.

Kapitola 5

Testování

Při testování je nutné brát v potaz i samotný generátor náhodných čísel. Ten se inicializuje funkcí `time (NULL)`; , která vrátí počet sekund uplynulých od 1. ledna 1970. Aby bylo každé generování nezávislé, je mezi nimi nutná minimální prodleva 1 sekunda, jinak by generátor dával stejné posloupnosti čísel. Tím by se v konečném důsledku generovaly stejné obvody.

Při testování se zaměřím pouze na kombinační obvody. Jak již bylo řečeno. Sekvenční obvod vzniká navýšením počtu vstupů a výstupů pro kombinační část a následném propojení obvodu typu D. Syntézni nástroje pracují při minimalizaci právě jen s kombinační částí. Z tohoto důvodu využiji tvorbu sekvenčních obvodů až při testování klonů standardních obvodů.

Pro testování klonů obvodů byly využity testovací benchmarky z MCNC [7], které obsahují obvody běžně používané v návrzích komplexních obvodů, jako jsou sčítačky, násobičky, posuvné registry atd. .

5.1 Testování rekonvergence

5.1.1 Závislost počtu smyček rekonvergenčí na parametrech obvodu

V první části testování se zaměříme na množství smyček rekonvergenčí, které vznikají při generování obvodů. Kompletní výsledky měření jsou na přiloženém CD souboru:

`data\rekonvergence\Rekonvergence1.xls`.

Parametry obvodů jsou vypsány v tabulce 5.1, všechny obvody měly stejný počet úrovní (20) a hradel (400), protože závislosti těchto parametrů jsou jasné. Při zvýšení počtu úrovní a hradel při náhodném generování dochází k zvýšení i počtu smyček rekonvergenčí. Budu proto měnit ostatní parametry, kde závislost není zřejmá.

Každé nastavení se změnilo vždy jen o jeden parametr vůči základu M1 s deseti vstupy, jedním výstupem, návazností jedna a maximálně dvěma vstupy hradel. Poté obdobně pro návaznosti signálů rovnou dvaceti v M5.

V tabulce je také vypsána hodnota nastavení šířky mřížky, kterou nastavuje program pomocí vstupních parametrů. Jedná se o omezení počtu hradel na jedné úrovni obvodu. Velké rozdíly jsou pouze při změně maximální návaznosti signálů. Při nastavení na hodnotu jedna, odvod velice rychle roste s velkým provázáním a nabírá hradla, proto musí být expanze omezena. Smyčky vznikají, ale

dosahují menší velikosti, jak ukazuje graf 5.1. Při největší návaznosti, dochází k opačnému jevu, šířka se musí rozšířit. Velký prostor s kombinací možnosti brát signály z jakékoli úrovně pozměnily smyčky v jejich velikosti. Došlo k nárůstu velikostí smyček, ale hlavní část dosahuje menších hodnot. Tento jev je zobrazen na grafu 5.2.

Měření	Vstupy	Úrovně	Hradel	Návaznost	Výstupy	Max. vst. H.	Šířka mřížky
M1	10	20	400	1	1	2	40
M2	10	20	400	1	5	2	38
M3	10	20	400	1	1	4	33
M4	20	20	400	1	1	2	40
M5	10	20	400	20	1	2	132
M6	10	20	400	20	5	2	130
M7	10	20	400	20	1	4	103
M8	20	20	400	20	1	2	132

Tabulka 5.1: Parametry obvodů rekonvergence

Měření	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Průměr
M1	1015	1001	981	969	969	1054	980	940	1013	985	991
M2	986	893	977	988	961	952	1014	860	1020	1043	969
M3	1328	1378	1530	1436	1316	1481	1433	1542	1474	1326	1424
M4	1130	1344	1157	1298	1134	1289	1283	1205	1375	1268	1248
M5	650	732	671	653	715	782	702	719	708	700	703
M6	681	658	720	669	665	714	736	687	695	734	696
M7	1034	1325	1140	1219	1278	1245	1217	1140	1127	1185	1191
M8	678	669	689	749	673	674	676	759	739	658	696

Tabulka 5.2: Počet rekonvergenčí

Výsledky deseti měření s každým nastavením jsou v tabulce 5.2 a ukazují na velký počet vzniklých základních smyček rekonvergenčí, které se při opakování měření mění jen minimálně, proto stačí i vzorek deseti měření. Jako referenční se bere nastavení M1 pro maximální návaznost vstupních signálů rovnou jedné a M5 pro neomezenou, tedy v tomto případě dvacet.

Průměrné množství základních rekonvergenčí se při nastavení M2 nepatrně snížilo. To je způsobeno zvýšením počtu výstupů z jednoho na pět. Všechny se nacházejí v poslední úrovni. Na vytvoření přídatných výstupů bylo potřeba čtrnácti hradel, které se přesunuly z těla obvodu na jeho konec, kde již jejich výstupní signály nevytvořily žádné další smyčky.

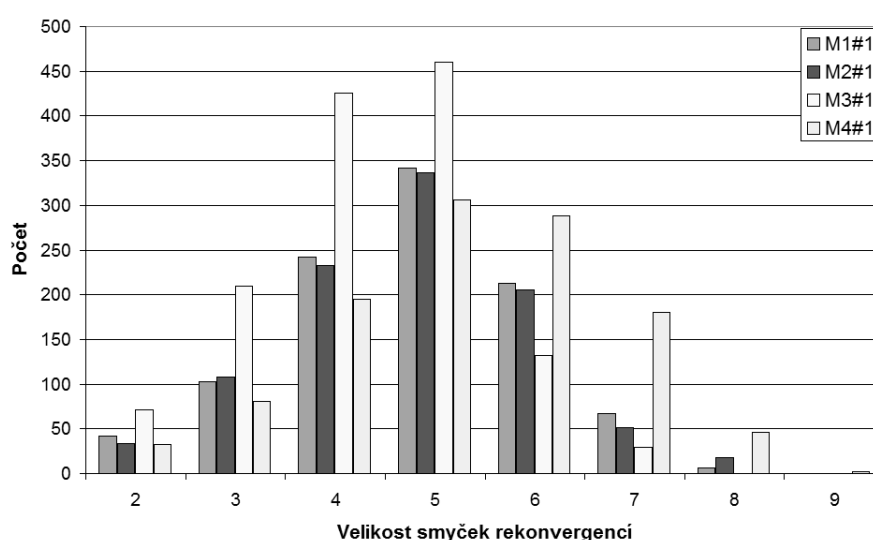
Při měření M3 došlo k nárůstu počtu smyček v průměru o 433 z důvodů zvýšení maximálního počtu vstupů hradla ze dvou na čtyři. Při zachování stejného počtu hradel se zvýšil počet vstupů a i počet větvení, což způsobilo onen nárůst.

Při nastavení M4 se průměrně zvýšila hodnota o 257 vůči referenční hodnotě, z důvodů zvětšení počtu vstupů obvodu na dvojnásobek. Větší množství vstupů tvoří v nulté úrovni více větvení signálů, které v konečném důsledku způsobí nárůst.

Druhé referenční měření je tentokrát s maximální návazností. Když se porovnají výsledky nastavení M1 a M5, je zřejmý výrazný pokles průměru počtu základních smyček rekonvergencí. Při možnosti získávat jeden vstupní signál z jakékoli minulé úrovně obvod hůře roste co do počtu hradel, proto se zde výrazně zvýšila nastavená šířka vnitřní mřížky hradel z 40 na 130.

Při porovnání M5 a M6 dojde k malému poklesu při zvýšení počtu výstupů na pět, což je obdobný jev jako při změně M1 a M2, pouze je pokles menší z důvodu použití menšího počtu hradel na dodatečné výstupy. Obdobná analogie se objevuje i při porovnání M5 a M7 s M1 a M3, tedy velký nárůst způsobený větším počtem vstupů hradla.

Zajímavý je až průměrný počet smyček při nastavení M8. Zde je patrný nevýrazný pokles oproti M5, což neodpovídá analogii M1 s M4, kde byl nárůst nepatrný. V tomto případě nemá zvýšení počtu vstupů výrazný vliv na počet smyček v obvodu.



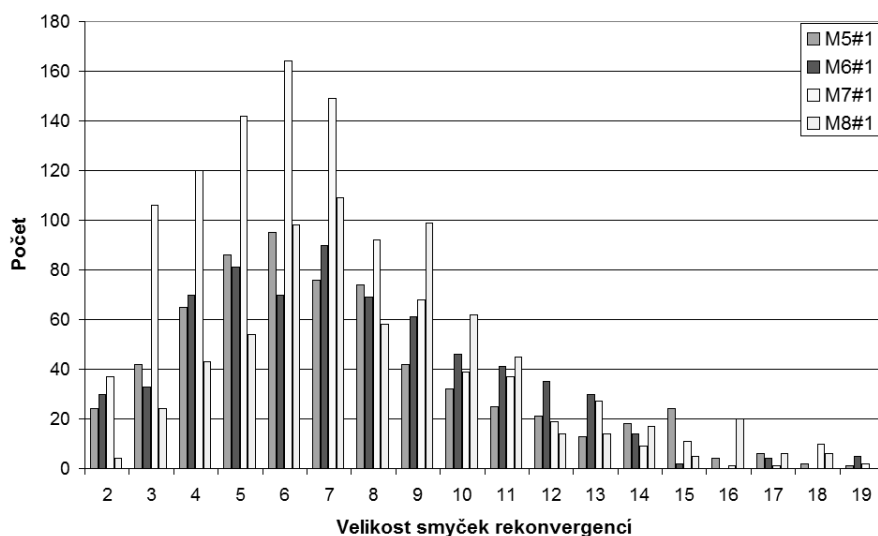
Obrázek 5.1: Graf rozložení velikosti rekonvergenčí pro nastavení M1 až M4

Na grafu 5.1, který je první k této problematice, se zobrazují velikosti základních smyček pro nastavení M1 až M4, jež jsou popsány v tabulce 5.1. Pro zobrazení bylo vždy vybráno první měření daného nastavení. Na první pohled je zřejmé, že při nastavení maximální návaznosti je velikost základních smyček maximálně devět úrovní a největší počet mělo velikost pět.

Při porovnání nastavení M1 a M2, došlo k poklesu počtu rekonvergencí na každé velikosti, zhruba o stejný díl, což odpovídá již zmíněným informacím tohoto porovnání výše.

Při zvýšení maximálního počtu vstupů hradla na čtyři, se nárůst projevil na nižších velikostech od dvou do pěti. Dále je počet rekonvergencí menší než u M1. Tento jev je způsoben tím, že dochází jak k většímu počtu větvení, tak i k většímu sdružování signálů, tím se velikost posouvá k menším hodnotám.

Poslední nastavení M4 zdvojnásobuje počet vstupních signálů a je jasně vidět posun velikosti k větším hodnotám. Při generování se každé hradlo snaží rozšířit počet závislých vstupů, To při zvýšení jejich počtu trvá více úrovní, a až poté se vytvoří větší počet rekonvergencí, které ale mají větší délku.



Obrázek 5.2: Graf rozložení velikosti rekonvergenčí pro nastavení M5 až M8

Graf 5.2 zobrazuje první naměřené hodnoty pro nastavení M5 až M8. Zvýšení maximální návaznosti se projevilo jak již zmiňovaným snížením počtu, ale také rozšířením rozsahu velikostí.

Při porovnání M5 a M6 není moc velký rozdíl jak v celkovém počtu, tak i v četnosti jednotlivých velikostí. Pouze došlo k malému posunu k vyšším velikostem smyček.

Nastavení M7 vykazuje velký kvantitativní nárůst ve všech velikostech, zvýšení počtu vstupů hradla velice ovlivní počet těchto smyček. Při posledním nastavení M8 dochází opět jako u M4 k velkému posunu velikostí smyček k vyšším hodnotám.

Měření ukázalo určité závislosti parametrů na změně počtu základních smyček rekonvergenčí. Jedná se hlavně o změnu počtu vstupů obvodu, kdy dojde k velkému nárůstu. Tento parametr jako jediný neovlivňuje nastavení šířky mřížky, takže výsledky nejsou nijak ovlivněny. Dalším výrazným parametrem je počet vstupů do logického prvku, kde je zaznamenán taktéž velký nárůst z důvodu zvýšení počtu větvení. Parametr maximální možné návaznosti signálů dosáhl vyšší hodnoty při nastavení 1, než při 20.

5.1.2 Projev snížení maximálního větvení signálu na počtu smyček rekonvergenčí

Smyčky rekonvergenčí vznikají při rozvětvení signálů. Generátor pracuje se dvěma parametry upravujícím větvení signálu. Průměrná hodnota větvení vychází z maximálního počtu vstupů hradla a ze zastoupení vícevstupých hradel v obvodu. Druhým parametrem je maximální možné větvení, na které se zaměříme v druhém testu rekonvergenčí.

Měření bude probíhat na dvojnásobných obvodech než měření předchozí. Počet vstupních signálů bude nastaven na 20, výstupních bude 10, úroveň 40, počet hradel 1000, maximální počet vstupů hradla bude nastaven na hodnotu 2. Měření proběhne pro tři různé hodnoty návaznosti vstupních signálů 1, 20 a 40. Pro každé nastavení proběhne generování padesáti obvodů, jednak pro snížené větvení, tak i pro neupravované. Parametr maximálního větvení bude nastaven na hodnotu jedna.

Toho nebude nikdy dosaženo, ale dojde k pokusu o snížení všech větvení, čímž se počet větvení rovnoměrně rozloží.

Kompletní výsledky se nacházejí na přiloženém CD v souboru
data\rekonvergence\Rekonvergence2.xls.

Návaznost:	1		20		40	
Snížení větvení:	ANO	NE	ANO	NE	ANO	NE
Průměr:	3,86	7,40	8,46	13,62	9,76	16,38
Min:	3	6	7	11	9	13
Max:	4	9	10	18	10	20

Tabulka 5.3: Průměrné snížení maximálního větvení signálů

Nejdříve se podíváme na vzniklé změny maximálního větvení signálu při použití algoritmu na snížení tohoto parametru. V tabulce 5.3 jsou výsledky z padesáti obvodů pro každé nastavení.

První řádek jsou průměrné hodnoty, následují dva řádky s minimálními a maximálními nalezenými hodnotami. Při generování obvodů s návazností 1 dochází k průměrnému maximálnímu větvení 7.4. Maximum dosahuje hodnoty 9 a minimum 6. Při snížení tohoto parametru klesne průměrná hodnota na 3.86, což je 52 % původní hodnoty. I hodnoty minima i maxima klesly na polovinu. Při nastavení s návazností 20 dochází k průměrnému počtu maximálního větvení 13.62. Při použití snižujícího algoritmu hodnota klesla na 8.46, což je 62 % původní hodnoty. Poslední nastavení dosahuje neupravovaná hodnota průměru 16.38. Bylo dosaženo snížení tohoto parametru na 9.76 při použití algoritmu. Hodnota je zde na 60 % původní. Algoritmus tedy funguje a nyní přistoupíme k zjištění změny počtu rekonvergenčí.

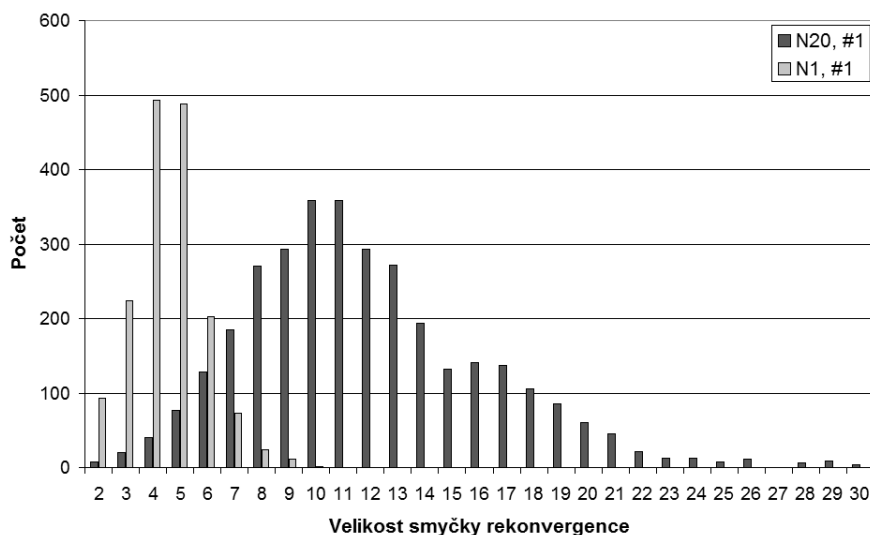
Návaznost:	1		20		40	
Snížení větvení:	ANO	NE	ANO	NE	ANO	NE
Průměr:	1738	1656	3220	3139	3069	2955
Min:	1581	1574	3031	2938	2757	2755
Max:	1841	1735	3431	3462	3265	3235

Tabulka 5.4: Závislost počtu rekonvergenčí na snížení větvení

V tabulce 5.4 jsou výsledky. První řádek je průměrný počet smyček rekonvergenčí z padesáti měření. Následující dva řádky, nalezené minimum a maximum z nalezených počtů. Při porovnání všech tří dvojic průměrných hodnot pro různá nastavení návaznosti je vidět velice malé zvýšení počtu rekonvergenčí při využití snížení maximálního větvení. Při pohledu na minimální a maximální hodnoty se tento rozdíl stírá.

Obvod s takto velkým počtem hradel vykazuje velké množství základních smyček rekonvergenčí. U předešlého měření byla závislost počtu smyček na návaznost signálů určena tak, že se zvýšením návaznosti z hodnoty jedna na maximální hodnotu sníží počet smyček. Zde tomu tak není. Při změně z hodnoty 1 na 20 dojde k téměř dvojnásobnému počtu smyček. Při opětovném zvýšení o 20 na maximální možnou hodnotu došlo k mírnému snížení, ale stále je tato hodnota daleko nad původním počtem smyček při návaznosti 1.

Tento posun lze odůvodnit nastavením šířky mřížky hradel v generovacím algoritmu. Šířka je vždy nastavena na co nejoptimálnější velikost pro vytvoření obvodu s hledaným počtem hradel. Při nastavení návaznosti na hodnotu 1 byla šířka stanovena na 37, a pro návaznost rovnou 40 na 103. Jinak by nedošlo k vytvoření hledaného obvodu. Malý počet smyček rekonvergenzí u návaznosti 1 je způsoben malou šířkou, kdy nebylo obvodu umožněno vytvořit více jak 37 hradel na jednu úroveň obvodu. Díky tomu vznikaly smyčky o menších velikostech a bylo jich méně.



Obrázek 5.3: Změna smyček rekonvergence při návaznosti 1 a 20

Jev je zobrazen na grafu 5.3 zobrazujícím četnost velikostí smyček. Je zde zobrazena první naměřená hodnota při návaznosti rovno 1 a 20, při nesnižování maximálního větvení.

5.1.3 Zhodnocení

Při náhodném generování obvodu vzniká velký počet smyček rekonvergenzí. Jejich počet nejvýrazněji ovlivňuje maximální počet vstupních signálů hradla, počet vstupních signálů a také zvolená maximální návaznost signálů. Dalším prvkem, který ovlivňuje vznik rekonvergenzí, je nastavení šířky generující mřížky. Jeho závislost však při měření nelze nijak odstínit, protože se nastaví na hodnotu vhodnou pro vytvoření požadovaného obvodu. V šírce je mezi měřeními velký rozdíl. Při nastavení parametru návaznosti signálů na hodnotu jedna dochází při generování obvodu k velkému zvyšování počtu hradel s přibývajícími úrovněmi. Pro tento případ se šířka nastaví na malou hodnotu, čímž omezí maximální počet hradel na jedné úrovni. Naopak při nastavení maximální návaznosti dochází k malým přírůstkům závislých hradel, tudíž se šířka mřížky hodně roztáhne. Tato skutečnost velice ovlivňuje měření, protože při změně jednoty parametru se změní dva.

K snižování maximálního větvení signálu dochází až po vygenerování obvodu, proto se na výsledcích žádná šířka mřížky neprojeví. Snižování větvení nijak neovlivní celkový počet signálů v obvodu, pouze rozprostře větvení po více signálech. Při měření došlo k mírnému nárůstu smyček rekonvergenzí při snížení maximálního větvení, avšak při pohledu na maximální a minimální dosažené

hodnoty se tento rozdíl ztrácí. Snížení maximálního větvení nijak neovlivní výsledný počet základních smyček rekonvergenčí vzniklých v obvodu.

5.2 Test vstupních parametrů v závislosti na minimalizovatelnosti

V druhé části této kapitoly provedeme testování schopnosti minimalizovat vygenerované obvody pomocí programu ABC [4]. Testy se budou týkat všech parametrů obvodu, díky tomu zjistíme závislost mezi parametry a schopností minimalizace.

5.2.1 Metodika

Celkově bude provedeno šest různých měření pro zjištění závislosti na počtu vstupů, výstupů, maximální návaznosti signálů, počtu vstupů logických prvků a rozdíl mezi obvody s a bez logických hradel XOR a NXOR. Poslední měření bude složeno z nejhorsí možné kombinace parametrů pro dosažení co nejhorsí minimalizace. Každý z těchto parametrů bude tvořen obvody o deseti, dvaceti a třiceti úrovních. Pro jednotlivý počet je vždy stejný počet hradel. Jedná se o 300, 800 a 1000. Aby bylo dosaženo získání přesných hodnot, bude vytvořeno vždy po deseti obvodech ke každému nastavení, jejichž hodnota po minimalizaci bude zprůměrována. Jednotlivé hodnoty jsou k nahlédnutí v příloze A od strany 88. Na příloženém CD ve složce `data\minimalizace` jsou kompletní výsledky tohoto měření.

Minimalizace bude probíhat pomocí nástroje ABC. Vnitřní algoritmus bude vždy desetkrát uveden v provoz pomocí posloupnosti příkazů `strash`, `dch` a `map`. Knihovna hradel načítaná tímto programem se vždy bude shodovat s parametrem maximálního počtu vstupů hradla, respektive množině možných vytvořených hradel.

Pro přehledné měření je důležité zvolit referenční parametry, z kterých budou jednotlivé testy vycházet. Tyto hodnoty jsem zvolil vzhledem k jejich středové poloze mezi nastaveními jednotlivých testů. Počet vstupů je 10, výstupů 5, maximální návaznost 10 a je vždy zvolena stejná četnost hradel. Jediný parametr, který je na možné hranici rozsahu, je maximální počet vstupů hradla. Základní hodnota je stanovena na dva vstupy.

Deset vstupů a pět výstupů se může zdát být málo. Tato hodnota byla však zvolena, aby co nejméně ovlivňovala výslednou strukturu obvodu. Obdobně je tomu i s hodnotou návaznosti. Počet deseti možných úrovní, ze kterých je možné získat vstupní signál, dává dostatečnou volnost. Všechny generované výstupy vždy vycházejí z nejvyšší úrovně.

Testování se týká pouze kombinačních obvodů. Sekvenční obvod by znamenal pouze zvýšení počtu vstupů a výstupů u kombinační části, se kterou se pracuje při minimalizaci.

5.2.2 Závislost na počtu vstupů

První měření se zabývá závislostí počtu vstupních signálů na celkové minimalizaci. Počty vstupů jsou 5, 10, 15 pro každý počet úrovní. V tabulce 5.5 jsou průměrné hodnoty počtu hradel po zminimalizování deseti obvodů, konkrétní výsledek jednotlivých obvodů obsahuje tabulka A.12. Také je zde změna velikosti, zapsaná v procentech. 100 % je původní velikost před minimalizací.

V tabulce je znázorněn výsledek. První ze sloupců je vždy původní hodnota před minimalizací a slouží k porovnání velikosti. Následující jsou již naměřené průměrné hodnoty. Nejnižší počet

Levels	Orig.	5PI	10PI	15PI	5PI [%]	10PI [%]	15PI [%]
20	300	236,8	275,2	284,4	78,9	91,7	94,8
30	800	660,9	769,2	773,2	82,6	96,2	96,7
40	1000	870,8	979,2	970,0	87,1	97,9	97,0

Tabulka 5.5: Průměrné hodnoty minimalizovaných obvodů, změna počtu vstupů

vstupů dosáhl i největší minimalizace. Od 79 % pro obvod s 20 úrovněmi do 87 % u 40 úrovní. Při zvýšení počtu vstupů na 10 se výrazně zvedlo i procento minimalizace. U 15 vstupů není nárůst již tak markantní. U největšího z obvodů došlo k nepatrnému poklesu, který lze přičíst kolísání průměru při dosažení horní hranice.

Počet vstupů obvodu není moc výrazný parametr co se týče minimalizace. Jeho největší vliv se projeví u menších obvodů, které jsou snadněji minimalizovatelné. U obvodu o 30 a 40 úrovních nedošlo při zvýšení z 10 na 15 k výraznější změně.

5.2.3 Závislost na počtu výstupů

Závislost na změně počtu výstupních signálů je měřena obdobně jako závislost vstupních. V tabulce 5.6 jsou průměrné počty hradel po minimalizaci spolu s procentuálním přepočtem. Kompletní výsledky jsou v tab. A.13.

Levels	Orig.	1PO	5PO	10PO	1PO [%]	5PO [%]	10PO [%]
20	300	281,0	275,2	278,1	93,7	91,7	92,7
30	800	753,5	769,2	770,5	94,2	96,2	96,3
40	1000	942,7	979,2	978,3	94,3	97,9	97,8

Tabulka 5.6: Průměrné hodnoty minimalizovaných obvodů, změna počtu výstupů

Z procentuálního zmenšení vyplývá prakticky neznatelná závislost. Je zde patrný malý několika procentní nárůst u obvodů s 30 a 40 úrovněmi při změně z 1 na 5 výstupů.

Závislost počtu výstupů není u větších obvodů, na kterých měříme, prakticky žádná.

5.2.4 Závislost na maximální návaznosti signálů

Třetí parametr pro zjištění míry minimalizace je maximální návaznost signálu. Pro všechny tři možnosti úrovní byly vytvořeny obvody s návazností 1, 10, 20 a pro obvody se 40 úrovněmi přibyla možnost délky signálů na 40. Tato hodnota omezuje pouze rozsah možností navázání vstupních signálů do hradla, nikde ale není stanoven průměr délky signálu. Tento parametr může být zavádějící v případech, kdy je maximální. Rovná se celkovému počtu úrovní obvodu.

V tabulce 5.7 vidíme průměrné hodnoty minimalizace, jednotlivé hodnoty jsou v tab. A.14. Dvačetiúrovňový obvod z původně 300 hradel dosáhl při návaznosti 1 necelých 91 %. Při zvýšení parametru stoupl poměr zmenšení o 1 %. Není patrný žádný rozdíl mezi návazností 10 a 20. Obvody o 30 úrovních dosáhly vyšších hodnot. Je patrné zvýšení při návaznosti 10, které je následováno poklesem o 3 % při návaznosti 20. I u největšího obvodu dojde mezi délkou 1 a 10 ke zvýšení z 94 % na

Levels	Orig.	1N	10N	20N	40N	1N [%]	10N [%]	20N [%]	40N [%]
20	300	272,3	275,2	275,1	–	90,8	91,7	91,7	–
30	800	732,6	769,2	745,1	–	91,6	96,2	93,1	–
40	1000	940,5	979,2	954,3	946,7	94,1	97,9	95,4	94,7

Tabulka 5.7: Průměrné hodnoty minimalizovaných obvodů, změna maximální návaznosti signálů

98 %. Následuje opět pokles o 2.5 % na hodnotu 95.4 % pro N = 20. Pokles pokračuje i v rozšíření na 40.

Návaznost signálů dosahuje nejmenší minimalizace kolem hodnoty 10.

5.2.5 Závislost na maximálním počtu vstupů hradla

Důležitým parametrem pro minimalizaci je použití vícevstupových hradel. Minimalizační algoritmus musí umět co nejvíce využít možností, které mu dávají hradla o více vstupech. Nyní se zaměříme na minimalizaci při použití hradel o maximálně 2, 3, nebo 4 vstupech.

Levels	Orig.	2HPI	3HPI	4HPI	2HPI [%]	3HPI [%]	4HPI [%]
20	300	275,2	399,1	431,4	91,7	133,0	143,8
30	800	769,2	1140,3	1253,6	96,2	142,5	156,7
40	1000	979,2	1535,1	1601,2	97,9	153,5	160,1

Tabulka 5.8: Průměrné hodnoty minimalizovaných obvodů, změna maximálního počtu vstupů hradla

V tabulce 5.8 jsou zobrazeny naměřené průměrné výsledky, z hodnot v tab. A.15. Při použití standardní dvouvstupové množiny logických prvků (2HPI) se ve všech případech pohybujeme pod hranicí 100 %. Zajímavá změna nastane při přechodu na třívstupová (3HPI). Zde došlo k nárůstu počtu hradel od jedné třetiny po celou polovinu. Pro obvody o 20 úrovních vzrostl koeficient na 133 %, pro 30 úrovní vzroste na 142,5 % a pro největší testovaný obvod na 153,5 %. Při použití knihovny s hradly i o 4 vstupech (4HPI), rostoucí trend pokračuje. Největšího zvětšení dojde v obvodech o 40 úrovních. Počet hradel se zde průměrně zvýší o 60 %, což je 600 hradel.

Při vytváření obvodů s hradly o více vstupech a jejich následné minimalizaci, i za použití adekvátní knihovny nedojde v žádném případě ke zmenšení daného obvodu.

5.2.6 Závislost na použití hradel XOR a XNOR

Logické hradlo XOR a jeho invertovaná verze tvoří pro nástroje zabývající se syntézou velké překážky. Poslední závislost objasní úspěšnost minimalizace při generování obvodů bez těchto hradel a porovná ji s obvody, které toto hradlo používají.

Naměřené hodnoty pro všech 10 obvodů všech nastavení jsou v tabulce A.16. Průměrné hodnoty jsou v tabulce 5.9. Při porovnání výsledných hodnot je patrné, že obvody bez hradel typu XOR se minimalizují lépe. Obvody s 300 hradly dosáhly 73 %, což je nejnižší hodnota naměřená v tomto testu. S přidáním možnosti vytvářet obvody se všemi typy hradel vzroste procentuální zmenšení na 92 %. Obdobně je to i u větších obvodů, kde se rozdíl pohybuje lehce pod 20 %.

Levels	Orig.	Bez X	S X	Bez XOR [%]	S XOR [%]
20	300	220,0	275,2	73,3	91,7
30	800	617,2	769,2	77,2	96,2
40	1000	813,9	979,2	81,4	97,9

Tabulka 5.9: Průměrné hodnoty minimalizovaných obvodů, použití hradel XOR a NXOR

Prokázalo se, že hradla XOR mají neblahý vliv na minimalizaci. V měřených případech dosahoval rozdíl průměrně 18 %.

5.2.7 Závislost na počtu vstupních signálů při nejhorší možnosti

Na závěr ze všech měřených parametrů vybereme ty, které co nejvíce negativně ovlivnily proces minimalizace. Opět budeme pracovat se třemi velikostmi obvodů a jako hlavní stupeň volnosti zvolíme počet vstupních signálů. Konstantní parametry jsou zřejmé z předchozích tabulek. Počet výstupních signálů bude roven 10, maximální návaznost je také 10 a jsou použita až čtyřvstupová hradla.

Levels	Orig.	5PI	10PI	15PI	5PI [%]	10PI [%]	15PI [%]
20	300	316,1	433,7	445,7	105,4	144,6	148,6
30	800	916,4	1234,6	1250,9	114,6	154,3	156,4
40	1000	1197,8	1582,3	1634,3	119,8	158,2	163,4

Tabulka 5.10: Průměrné hodnoty minimalizovaných obvodů, nejhorší případ

V tabulce 5.10 jsou zobrazeny získané průměrné hodnoty. Ani s pouze pěti vstupy nebylo dosaženo minimalizace. Hodnota je zde 105 %. Při nárůstu počtu vstupů u všech velikostí obvodu dochází k nárůstu, který však při zvýšení na 15 vstupů není již tak výrazný. Obdobné výsledky ukázalo i první měření závislosti na počtu vstupů. Maximální možnosti nebo spíše nemožnosti bylo dosaženo v obvodech o 30 úrovních, 1000 hradlech a 15 vstupech. Hodnota zde dosáhla 163,4 %.

5.2.8 Zhodnocení

Výsledky měření závislosti vstupních parametrů na minimalizaci pomocí nástroje ABC ukázaly na velký problém s hradly XOR při minimalizaci. Program překvapil neschopností zmenšit obvody vygenerované s tří a vícevstupovými hradly. Zde došlo vždy ke zvětšení obvodu i přestože program ABC pracoval se stejnou množinou hradel. Využití celé množiny nástrojem potvrdila analýza minimalizovaných obvodů.

Při zvyšování počtu vstupních signálů docházelo i k nárůstu poměru mezi původní velikostí a zmenšeným obvodem. Rozdíl mezi deseti a patnácti vstupy již nebyl tak výrazný. Obdobně, pouze s menší závislostí, se tento jev objevil i u počtu výstupních signálů. Tyto parametry nemají u velkých souborů velký vliv. Maximální návaznost vstupních signálů hradla ovlivnilo minimalizaci také, ale pouze v rámci několika procent. Rozhodujícími parametry jsou již zmiňovaná hradla XOR spolu s využitím vícevstupových hradel. Při nastavení nejhorší možné situace došlo u největšího obvodu k průměrnému nárůstu počtu logických prvků o 63 %.

Měření probíhalo na obvodech třech velikostí. První dosahoval dvaceti úrovní a tři sta hradel, druhý třiceti úrovní a osmi sty hradel, třetí měl čtyřicet úrovní a tisíc hradel. Je velice obtížné zjišťovat závislost parametrů z důvodů vnitřní struktury obvodu. Pro různá nastavení počtu úrovní, počet výstupů, návaznosti signálů, počtu vstupů hradel a i počtu logických prvků se mění šířka vnitřní mřížky hradel. To znamená, že se změni omezení maximálního počtu logických prvků na jedné úrovni obvodu. Proto při změnách nastavení nedojde pouze k jedné změně, právě zamýšlené hodnoty, ale také k různému omezení obvodu rozrůstat se do šířky. V měření bylo s generátorem pracováno jako s černou skříňkou. Zjištěné hodnoty pro změnu počtu vstupů a použití hradel XOR platí, protože tyto parametry jako jediné neovlivňují přímo nebo nepřímo šířku mřížky. U ostatních měření může dojít k různým odchylkám při práci v jiném nastavení. Tento fakt je dán principem generování a není možné jeho vlivu zamezit.

5.3 Klony standardních zkušebních obvodů

V poslední části této kapitoly se zaměříme na vytváření souborů podle parametrů získaných z již existujícího obvodu. Jako vzorové obvody poslouží množina obvodů z balíku MCNC [7].

5.3.1 Metodika

Vybraný obvod se nejdříve přemapuje pomocí syntetizačního nástroje ABC pomocí knihovny hradel `5g_all.genlib` na obvod skládající se ze základní množiny hradel. Přemapování způsobí pouze čistý převod ze souboru `blif`, který nemá nějak specifikované či standardizované logické bloky, na soubor standardních bloků ve formátu pro načtení programem. V knihovně `5g_all.genlib` se nacházejí logické prvky s maximálně pěti vstupy. Tato hodnota je volena s ohledem na co minimální omezení počtu vstupů a zároveň zamezí sporadickému vytváření hradel s velkým počtem vstupů. Takto upravený obvod se načte pomocí programu a získá se počet vstupů, počet výstupů a jejich úrovně, počet hradel, maximální návaznost signálů, průměrné a maximální větvení a také počet jednotlivých logických prvků. V případě sekvenčních obvodů se samozřejmě získá počet klopných obvodů a počet úrovní logiky mezi vstupem a výstupem jednotlivého klopného obvodu. Tyto parametry uloží ve formátu pro generování obvodu, díky tomu je možné vytvářet klony. Získáme také doplňující informaci o počtu základních smyček rekonvergenčí.

Pro každý obvod bylo vždy vytvořeno deset klonů, které se následně podrobily minimalizaci nástrojem ABC spolu s původním přemapovaným obvodem. Každá minimalizace se opakovala desetkrát a až poté byl uložen soubor, který se programem analyzoval. Následně se porovnály parametry původního přemapovaného obvodu s jeho klony a svými minimalizovanými protějšky.

Ne pro každý obvod ze skupiny MCNC lze vytvořit přesné klony. Důvod pochází jak ze samotného principu generování, kdy je požadavek na vytvoření jednovýstupového obvodu, který je závislý na všech vstupech. Obvod s velkým množstvím vstupů a výstupů, ale s malým počtem úrovní, nelze vytvořit. Druhý důvod pochází z přesného nastavení požadavků na četnost jednotlivých typů hradel. Při velké preferenci určitého typu může dojít ke změně struktury v mřížce hradel, čímž nedojde k nalezení potřebného obvodu.

5.3.2 Obvod apex6.blif

První kombinační obvod pro vytvoření klonů je apex6.blif. Obvod po přemapování na maximálně pětivstupá hradla obsahuje 598 prvků. Tento počet neobsahuje, jako u všech ostatních měření, buffery a konstantní signály vložené v obvodu. Obvod je členěn do dvanácti úrovní s velkým počtem vstupů (135) a výstupů (99) rozprostřených po všech úrovních. Tyto parametry budou vždy konstantní pro všechny vytvořené klony tohoto obvodu a přehledně jsou zobrazeny v tabulce 5.11 .

Vstupy	Výstupy	Úrovně	Hradla
135	99	12	598

Tabulka 5.11: Stejně parametry obvodu apex6.blif s klony

V následující tabulce 5.12 jsou zobrazeny ostatní parametry obvodu a jeho klonů. Maximální návaznost signálů (Nav.) u originálu dosahuje hodnoty 11, což se u klonů ne vždy podařilo dodržet. Minimální délka je 9, tedy se velmi přibližuje originálu. U parametru maximálního větvení (M.v.) signálu jsou rozdíly větší. Originál dosahuje hodnoty 25 oproti ani ne poloviční hodnoty 11 u klonů. Algoritmy generátoru nemají žádnou možnost větvení navýšit, mohou ho pouze snižovat. Druhým parametrem větvení je jeho průměr (Pr.v.). Originál dosahuje hodnot 1.84, zatím co ostatní obvody se pohybují kolem hodnoty 2.00. Poslední hodnoty, které jsou obsaženy v generujících parametrech jsou počty jednotlivých typů hradel. Obvod apex6.blif obsahuje téměř z 77 % logické prvky NAND. Kopie dosahují vyšších hodnot, ale velice dobře dodržují poměrné množství všech typů hradel. Poslední hodnotou v tabulce je počet smyček rekonvergence (Rek.). Ta sice není obsažena v generujících parametrech, ale zde slouží k hrubému porovnání vnitřní struktury obvodu. v původním obvodu jich bylo nalezeno 521, což je méně než u všech klonů. Jediná výjimka je u klonu K5, kde je hodnota 489.

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	11	25	1,84	51	2	4	30	6	460	45	521
K1	10	11	2,02	43	7	10	22	3	502	11	767
K2	10	8	2,01	51	5	8	20	4	487	23	808
K3	9	9	1,99	49	2	10	23	1	493	20	767
K4	10	9	2,00	47	3	9	18	4	505	12	685
K5	9	8	1,98	54	3	5	32	1	489	14	489
K6	10	9	2,03	50	2	11	20	1	495	19	709
K7	10	9	2,04	51	2	5	27	4	492	17	869
K8	10	9	2,02	46	2	9	27	3	497	14	737
K9	11	11	2,00	54	8	7	20	5	492	12	685
K10	11	11	2,05	39	3	4	17	3	514	18	787

Tabulka 5.12: Parametry obvodu apex6.blif a jeho klonů

Výsledky po minimalizaci obvodů jsou zobrazeny v tabulce 5.13 a jsou zajímavé. V atributu počtu prvků došlo k nejvyšší minimalizaci právě u původního souboru a dosahovala hodnoty 90 %. Klony obvodu dosahovaly minimalizace v rozmezí 96 až 103 %, v několika případech se obvod

i zvětšil. Této skutečnosti odpovídá i počet rekonvergenčí, který většinou vzrostl. Ostatní parametry se minimalizací prakticky nijak nezměnily.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	10	540	9	22	1,84	494	90,30
K1	12	592	10	9	2,05	852	99,00
K2	11	601	9	8	2,05	1122	100,50
K3	12	597	10	10	2,01	753	99,83
K4	11	618	9	9	2,03	739	103,34
K5	10	574	10	8	2,04	501	95,99
K6	12	585	10	9	2,07	719	97,83
K7	12	595	10	9	2,08	851	99,50
K8	12	588	9	9	2,05	747	98,33
K9	13	609	11	11	2,01	718	101,84
K10	12	613	11	11	2,05	802	102,51

Tabulka 5.13: Parametry po minimalizaci souboru apex6.blif a jeho klonů

Testovacímu obvodu byly vytvořeny velice podobné klony, v rámci měřených parametrů, které poukázaly na špatnou možnost minimalizace. Podobné obvody dokonce dosáhly horší minimalizace než samotný originál.

5.3.3 Obvod apex7.blif

Jako druhý testovaný obvod byl zvolen apex7.blif. Základní parametry obvodu jsou zobrazeny v tabulce 5.14. Počet vstupních signálů je 49, výstupních 37 a počet úrovní obvodu po přemapování na pětivstupá hradla dosahuje hodnoty 14. Počet prvků je nyní 216.

Vstupy	Výstupy	Úrovně	Hradla
49	37	14	216

Tabulka 5.14: Stejně parametry obvodu apex7.blif s klony

Ostatní parametry zobrazené v tabulce 5.15 se týkají jak originálu, tak vygenerovaných deseti klonů. Maximální návaznost 12 je nyní dosažena téměř v naprosté většině případů. Ostatní hodnoty jsou svými vlastnostmi obdobné jako u předešlého souboru apex6.blif. Maximální větvení signálu dosahuje u předlohy hodnoty 26, zatím co u klonů se tato hodnota pohybuje pod polovinou originálu. I průměrné větvení se pohybuje mírně nad vzorovou hodnotou 1.79. Skladba typů je poměrově stejná jako u předchozího obvodu, opět se zde preferují hradla typu NAND. Rekonvergenčí je díky menšímu počtu hradel méně, 213 u originálu. Klony obvodu dosahují výrazně většího počtu.

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	12	26	1,79	32	1	0	5	5	151	22	213
K1	12	7	1,98	28	0	0	4	1	169	14	289
K2	12	7	1,97	33	2	0	1	2	167	11	341
K3	12	9	2,00	38	1	0	5	4	153	15	407
K4	12	9	1,93	33	0	0	5	5	168	5	262
K5	12	9	1,94	29	3	0	5	6	165	8	269
K6	12	8	1,95	31	3	0	4	1	169	8	281
K7	12	6	1,90	29	4	0	4	2	169	8	252
K8	12	9	1,94	32	1	0	4	3	168	8	307
K9	12	8	1,91	42	2	0	8	3	158	3	261
K10	11	9	1,94	37	0	0	6	2	157	14	234

Tabulka 5.15: Parametry obvodu apex7.blif a jeho klonů

Nyní se podíváme do tabulky 5.16, kde je přehledně zobrazen výsledek minimalizace. Největší pokles počtu hradel je opět u originálního obvodu 73 %. Nejvíce zmenšený obvod z klonů dosahuje 90 % u K10. Naopak K7 nebyl s 99 % téměř zmenšen. U rekonvergenčí došlo k pohybu hodnoty na obě strany. Zminimalizovaný originál má největší úbytek smyček.

Obvod apex7.blif byl velice dobře zduplikován generátorem. Originál dosahuje největšího procenta zmenšení jakož i nejmenšího počtu smyček rekonvergence. Tato závislost ale není potvrzena, protože klony se minimalizovaly nezávisle na kladném, či záporném přírůstku smyček.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	11	159	10	20	1,80	152	73,61
K1	9	212	9	7	2,02	317	98,15
K2	13	203	12	7	2,02	329	93,98
K3	11	209	10	7	2,02	422	96,76
K4	11	195	10	9	2,01	251	90,28
K5	10	206	9	10	1,98	267	95,37
K6	9	201	8	8	1,96	212	93,06
K7	9	213	8	8	1,97	323	98,61
K8	10	204	10	9	2,04	337	94,44
K9	12	194	11	7	1,99	259	89,81
K10	8	194	8	8	2,03	196	89,81

Tabulka 5.16: Parametry po minimalizaci souboru apex7.blif a jeho klonů

5.3.4 Obvod b9.blif

Obvod b9.blif patří mezi menší obvody. Má pouze 89 hradel. Počet vstupů a výstupů dosahují hodnot 41 a 21. Vše je tvořeno v devíti úrovních obvodu. Tyto neměnné parametry jsou v tabulce 5.17.

Vstupy	Výstupy	Úrovně	Hradla
41	21	9	89

Tabulka 5.17: Stejně parametry obvodu b9.blif s klony

Jako u všech testovaných obvodů zde bylo vytvořeno 10 klonů, parametry jsou v tabulce 5.18. Všechny mají stejnou velikost návaznosti signálů společnou také s originálem. Návaznost má hodnotu 7. Maximálního větvení nebylo u klonů dosaženo, ale tentokrát jsme se přiblížili hodnotě průměrného větvení originálu 1.59. Poměr mezi jednotlivými typy hradel odpovídá u klonů originálu. Naměřená hodnota rekonvergence se zvýšila z počtu 31 u originálu na maximálně 52 u klonu K6.

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	7	11	1,59	21	0	0	4	4	49	11	31
K1	7	6	1,64	14	0	0	4	4	57	10	50
K2	7	4	1,62	19	0	0	4	4	55	7	35
K3	7	6	1,68	17	0	0	2	6	56	8	48
K4	7	6	1,65	13	0	0	5	3	63	5	33
K5	7	6	1,67	14	0	0	4	3	54	14	40
K6	7	5	1,72	27	0	0	8	4	61	9	52
K7	7	5	1,63	19	0	0	4	4	44	18	36
K8	7	5	1,60	15	0	0	3	4	63	4	35
K9	7	5	1,58	22	0	0	4	2	57	4	43
K10	7	4	1,66	17	0	0	2	3	60	7	45

Tabulka 5.18: Parametry obvodu b9.blif a jeho klonů

Po minimalizaci zobrazené v tabulce 5.19 došlo ke snížení počtu úrovní z devíti na sedm a méně. Originál dosáhl zmenšení 90 %, což je průměrná hodnota dosažená u klonů. K9 dosáhl také s počtem sedmi úrovní 81 %. Naopak K6 dosáhl poměrného zmenšení 107 %. Asi nejpodobnějším klonem, co se týče parametrů, je K8. I počet rekonvergenzí je velice blízký originálu. K8 se i podobně choval při minimalizaci a stejného počtu hradel jako zminimalizovaný originál.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	7	80	5	9	1,54	35	89,89
K1	5	85	5	7	1,75	54	95,51
K2	6	79	6	5	1,70	35	88,76
K3	7	82	6	6	1,75	52	92,13
K4	7	80	6	6	1,68	30	89,89
K5	6	90	6	6	1,66	40	101,12
K6	6	96	6	5	1,74	41	107,87
K7	6	79	5	5	1,69	41	88,76
K8	6	80	6	5	1,65	32	89,89
K9	7	72	5	5	1,63	36	80,90
K10	6	83	6	4	1,71	47	93,26

Tabulka 5.19: Parametry po minimalizaci souboru b9.blif a jeho klonů

5.3.5 Obvod c880.blif

Obvod c880.blif z testovací sady má 236 hradel, 60 vstupů a 14 úrovní. Vše je zaznamenáno v tabulce 5.20. Klony i samotný obvod mají návaznost signálů rovnou devíti. U průměrného větvení bylo dosaženo stejné hodnoty 1,96 u všech obvodů i maximální větvení se příliš nelišilo od originálu. Poměr typů hradla se s určitou tolerancí shoduje. Na rozdíl od předešlých testů má zde původní obvod mnohem více rekonvergenčí než jeho klony. Bylo naměřeno 909 smyček oproti 697 smyčkám u klonu K8. Výsledky jsou v tabulce 5.21.

Vstupy	Výstupy	Úrovně	Hradla
60	26	14	236

Tabulka 5.20: Stejně parametry obvodu c880.blif s klony

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	9	9	1,96	21	18	0	19	10	149	19	909
K1	9	7	1,96	20	30	0	16	6	157	7	514
K2	9	8	1,96	22	26	0	17	8	148	15	548
K3	9	8	1,96	29	35	0	8	3	146	15	626
K4	9	7	1,96	16	26	0	17	6	161	10	566
K5	9	8	1,96	25	29	0	14	11	150	7	489
K6	9	6	1,96	27	31	0	10	4	155	9	636
K7	9	7	1,96	25	26	0	17	9	149	10	627
K8	9	6	1,96	21	30	0	12	2	161	10	687
K9	9	8	1,96	21	31	0	19	9	145	11	544
K10	9	7	1,96	27	28	0	13	3	148	17	533

Tabulka 5.21: Parametry obvodu c880.blif a jeho klonů

Parametry minimalizovaných obvodů z tabulky 5.22 ukazují, že originální obvod c880 se nepodařilo zminimalizovat. Má stejný počet hradel, došlo však ke zvýšení počtu úrovní o tři a nastal pokles počtu rekonvergenčí. U klonů došlo k zvětšení počtu hradel tak i úrovní. Velký rozdíl nastal v počtu rekonvergenčí, kde se navzdory tendenci originálu ke zmenšení, jejich počet markantně zvýšil.

U tohoto obvodu poprvé minimalizační algoritmus neuspěl. Originální obvod se zmenšil pouze v počtu rekonvergenčí, ale počet logických prvků byl zachován. U klonů došlo k nárůstu počtu prvků a také rekonvergenčí.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	12	236	11	11	1,95	794	100,00
K1	13	259	9	6	1,95	535	109,75
K2	12	263	8	9	2,02	659	111,44
K3	13	267	10	9	2,11	971	113,14
K4	13	251	9	8	2,05	717	106,36
K5	12	275	9	8	2,03	848	116,53
K6	14	266	9	7	1,98	795	112,71
K7	14	245	12	8	2,05	681	103,81
K8	13	272	9	7	2,04	1026	115,25
K9	13	253	11	8	2,10	839	107,20
K10	11	243	10	8	2,08	658	102,97

Tabulka 5.22: Parametry po minimalizaci souboru c880.blif a jeho klonů

5.3.6 Obvod c1908.blif

Obvod c1908.blif se skládá z 33 vstupů, 25 výstupů, 17 úrovní a 237 hradel. Jedná se tedy o relativně malý obvod vůči předchozím. Parametry jsou zobrazeny v tabulce 5.23 .

Vstupy	Výstupy	Úrovně	Hradla
33	25	17	237

Tabulka 5.23: Stejně parametry obvodu c1908.blif s klony

Tvorba deseti klonů proběhla bez jakýchkoliv problémů a jejich parametry jsou v tab. 5.24. Maximální návaznost originálů byla jen s minimálním rozptylem splněna. Většího odchylení došlo u parametru maximálního větvení, kde klon K5 dosáhl hodnoty 16, což je o pět více než originál. Ostatní klony tento parametr dodržely. Průměrné větvení originálu bylo o jednu desetinu větší než hodnota, kolem které se pohybovaly duplikáty. Zajímavá je hodnota o počtu základních smyček rekonvergenčí, kde originál dosáhl počtu 1554. Nejbliže k této hodnotě je obvod K7 s pouze 524.

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	15	11	2,03	29	52	22	35	3	80	16	1554
K1	15	11	1,93	29	57	43	26	1	69	12	452
K2	15	11	1,87	43	56	38	19	1	69	11	477
K3	15	8	1,89	25	88	30	26	2	57	9	432
K4	14	10	1,83	32	86	46	19	3	44	7	422
K5	14	16	1,90	29	75	35	19	4	62	13	447
K6	15	10	1,88	29	84	36	24	0	51	13	434
K7	15	11	1,88	28	84	32	20	1	63	9	524
K8	15	11	1,87	30	77	38	22	0	57	13	446
K9	14	10	1,89	41	68	28	25	4	53	18	402
K10	15	9	1,89	30	92	25	17	1	61	11	419

Tabulka 5.24: Parametry obvodu c1908.blif a jeho klonů

Výsledky minimalizace z tab. 5.24 ukazují na všeobecný nárůst hradel u všech obvodů. Minimalizovaný originál zvýšil svojí velikost o 37 %. Nejbliže k této hodnotě byl obvod K7 s 149 %. Právě tento klon dosahoval před minimalizací největšího počtu rekonvergenčí. K5, který se vymykal v parametru maximálního větvení se zvětšil o 69 %. Zajímavá je i informace o počtu rekonvergenčí, kde originál dosahuje počtu 2952. K10 mu šlape na paty s 1910, ale zvětšením o 63 % svůj vzor překonal.

Výsledky dosažené tímto testovacím obvodem jsou zajímavé. Při co největším dodržení parametru obvodu nebylo dosaženo tolika rekonvergenčí. Po minimalizaci však měly všechny obvody velkou tendenci ke zvětšení.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	16	348	15	18	2,36	2952	146,84
K1	17	339	14	13	2,14	1245	143,04
K2	14	329	13	9	2,17	1534	138,82
K3	14	355	12	10	2,19	1890	149,79
K4	15	340	13	9	2,12	1398	143,46
K5	14	401	12	13	2,27	1640	169,20
K6	16	329	15	11	2,15	1627	138,82
K7	17	352	15	11	2,10	1603	148,52
K8	15	321	13	9	2,12	1410	135,44
K9	16	326	13	9	2,09	1141	137,55
K10	15	387	14	10	2,25	1910	163,29

Tabulka 5.25: Parametry po minimalizaci souboru c1908.blif a jeho klonů

5.3.7 Obvod count.blif

Parametry obvodu count.blif jsou zobrazeny v tab. 5.26, proto je dále nebudu popisovat. Důležitá je až tab. 5.27 s hodnotami, které se mohou lišit. Maximální návaznost je stejná u všech obvodů, pouze jeden má tento parametr o jedna menší. Velký rozdíl je vidět u maximálního větvení signálu, kde originál má počet 32 a klony dosahují maximálně na třetinu této hodnoty. U obvodů se 116 hradly je tak velké větvení neobvyklé. Do bezmála třetiny všech prvků vchází jeden signál. Hodnota průměrného větvení je u všech obvodů stejná.

Vstupy	Výstupy	Úrovně	Hradla
35	16	8	116

Tabulka 5.26: Stejně parametry obvodu count.blif s klony

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	7	32	1,87	19	0	0	5	22	65	5	116
K1	7	9	1,87	19	0	0	5	19	68	5	162
K2	7	10	1,87	12	0	0	7	17	73	7	161
K3	7	9	1,87	18	0	0	2	13	83	0	207
K4	7	7	1,87	15	0	0	4	21	76	0	165
K5	7	8	1,84	22	0	0	3	14	71	6	175
K6	6	7	1,87	18	0	0	4	13	78	3	164
K7	7	7	1,87	16	0	0	4	20	74	2	197
K8	7	8	1,87	15	0	0	3	16	79	3	158
K9	7	9	1,87	17	0	0	3	17	77	2	190
K10	7	8	1,81	24	0	0	3	16	71	2	177

Tabulka 5.27: Parametry obvodu count.blif a jeho klonů

Minimalizované klony z tab. 5.28 dosahují procentuálního zmenšení od 83 % do 102 %. Původní obvod má tento poměr o málo vyšší konkrétně 105 %. Zajímavý je počet rekonvergenčí, který se u originálu snížil, kdežto u klonu většinou mírně zvýšil.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	6	122	5	32	2,04	76	105,17
K1	7	108	7	8	1,88	177	93,10
K2	6	111	6	9	1,90	138	95,69
K3	9	118	8	8	1,88	209	101,72
K4	8	105	7	6	1,90	209	90,52
K5	7	107	7	7	1,89	228	92,24
K6	7	96	6	7	1,95	176	82,76
K7	8	114	7	7	1,86	198	98,28
K8	8	118	7	8	1,84	164	101,72
K9	8	110	8	8	1,88	209	94,83
K10	9	105	8	8	1,86	199	90,52

Tabulka 5.28: Parametry po minimalizaci souboru count.blif a jeho klonů

5.3.8 Obvod term1.blif

Stabilní parametry jsou v tabulce 5.29. Obvod term1.blif nemá nikterak výrazné parametry. V tabulce 5.30 je vidět atributy klonů, které jsou prakticky totožné s přemapovaným originálem. Počet smyček rekonvergenčí je 287, což je průměrně o 21 méně než u klonů. Přesto lze považovat i tuto hodnotu za dodržanou.

Vstupy	Výstupy	Úrovně	Hradla
34	10	10	195

Tabulka 5.29: Stejně parametry obvodu term1.blif s klony

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	9	11	2,00	31	0	0	21	9	119	15	287
K1	8	10	2,00	38	0	0	13	11	126	7	328
K2	8	10	2,00	47	0	0	14	7	121	6	295
K3	8	10	2,00	38	0	0	13	4	136	4	333
K4	8	11	2,00	39	0	0	13	7	132	4	295
K5	9	11	1,98	43	0	0	16	12	119	5	298
K6	8	9	2,00	41	0	0	16	6	129	3	333
K7	9	11	2,00	41	0	0	16	11	117	10	330
K8	8	10	2,00	37	0	0	21	4	126	7	299
K9	8	11	2,00	42	0	0	11	4	133	5	308
K10	9	11	2,00	37	0	0	18	7	127	6	261

Tabulka 5.30: Parametry obvodu term1.blif a jeho klonů

U minimalizovaných obvodů, jejichž hodnoty jsou v tab. 5.31, je zřejmý největší úbytek hradel u originálu konkrétně na 54 %. Klony se pohybovaly v rozmezí 79 % až 92 %, což jsou vyšší hodnoty než u originálu. Za zmínku stojí i snížení počtu smyček obvodu o 167 u originálu. U klonů nedošlo k tak výrazné změně. Hodnoty se zde pohybují od úbytku 50 smyček u K9 po zvýšení o 55 u K5.

U tohoto obvodu bylo dosaženo přesného vytvoření 10 klonů, i přesto nedošlo ke stejnému stupni minimalizace jako u originálu. Ukazuje se zde vlastnost generovaných obvodů velice účinně vzdorovat minimalizačním nástrojům.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	8	106	6	7	1,73	120	54,36
K1	9	179	7	9	2,04	354	91,79
K2	9	169	7	10	2,11	342	86,67
K3	8	155	6	10	2,10	382	79,49
K4	11	178	8	9	2,04	298	91,28
K5	9	156	8	9	2,12	353	80,00
K6	9	169	7	9	2,10	310	86,67
K7	10	167	7	9	2,09	335	85,64
K8	9	168	8	11	2,12	340	86,15
K9	8	159	7	10	2,13	258	81,54
K10	9	164	9	11	2,06	229	84,10

Tabulka 5.31: Parametry po minimalizaci souboru term1.blif a jeho klonů

5.3.9 Obvod too-large.blif

Posledním testovaným kombinačním obvodem je too-large.blif. Jedná se o obvod s 38 vstupy, 3 výstupy, 459 hradly, které se rozprostírají na 25 úrovních. Tyto neměnné parametry jsou v tab. 5.32. V následující tabulce 5.33 jsou parametry všech klonů tohoto obvodu. V obvodu je výrazná maximální návaznost signálů, která je u originálu 23 úrovní. Klony se této hodnotě dobře přibližují, což se nedá říci o maximálním větvení signálu, kde byla dosažena jen polovina požadované hodnoty 24. Parametry se na rozdíl v průměrném větvení 2.15 přesně shodují. Poměrné rozdělení typů hradel je již tradičně shodné s originálem. Rozdíl ve vnitřní struktuře napovídá různý počet smyček rekonvergenčí. U originálu bylo napočítáno 1263 oproti klonům, kde se počet pohyboval v rozmezí 1813 u K5 po 2090 u K1.

Vstupy	Výstupy	Úrovně	Hradla
38	3	25	459

Tabulka 5.32: Stejně parametry obvodu too-large.blif s klony

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	23	24	2,15	46	0	0	24	4	327	58	1263
K1	19	11	2,15	60	0	0	28	2	357	12	2090
K2	23	10	2,15	59	0	0	22	2	356	20	2034
K3	22	12	2,15	56	0	0	20	2	365	16	2057
K4	22	11	2,15	53	0	0	18	2	365	21	1904
K5	22	13	2,15	50	0	0	20	2	361	26	1813
K6	21	9	2,15	49	0	0	15	2	377	16	1980
K7	22	10	2,15	60	0	0	13	4	362	20	2076
K8	20	9	2,15	61	0	0	22	1	358	17	2080
K9	17	10	2,15	65	0	0	18	3	360	13	1961
K10	22	10	2,15	59	0	0	24	0	352	24	1969

Tabulka 5.33: Parametry obvodu too-large.blif a jeho klonů

Proces minimalizace opakovaný desetkrát syntetizačním nástrojem ABC provedl na testovaných obvodech změny, které jsou zobrazené v tabulce 5.34. Originální obvod dosáhl 59 % své původní velikosti. Klony obvodu too-large.blif. byly odolnější. Hodnoty se pohybují mezi 88 % až 95 %. Opět je zde patrný jev, kdy u originálního obvodu klesne počet smyček rekonvergenčí (z 1263 na 529), zatímco klony, ba i se stejnými parametry, prakticky nezmění počet smyček.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	15	271	12	11	1,91	529	59,04
K1	19	413	15	10	2,24	1994	89,98
K2	22	423	20	10	2,23	2017	92,16
K3	18	420	16	12	2,26	2144	91,50
K4	22	428	19	10	2,22	1925	93,25
K5	18	403	17	11	2,22	1734	87,80
K6	16	435	14	11	2,22	1974	94,77
K7	23	416	21	10	2,26	2202	90,63
K8	20	425	16	10	2,23	2117	92,59
K9	18	409	13	9	2,27	1971	89,11
K10	18	417	16	10	2,25	1985	90,85

Tabulka 5.34: Parametry po minimalizaci souboru too-large.blif a jeho klonů

5.3.10 Obvod dk15.blif

První sekvenční obvod, kterému se budou vytvářet klony je dk15.blif. V tabulce 5.35 jsou zobrazeny parametry společné pro všechny obvody. Tento testovací obvod je celkem malý, má pouze 3 vstupy a 5 výstupů. 36 hradel je uspořádáno ve 4 úrovních. Nacházejí se zde 2 klopné obvody.

Vstupy	Výstupy	Latch	Úrovně	Hradla
3	5	2	4	36

Tabulka 5.35: Stejně parametry obvodu dk15.blif s klony

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	2	11	3,02	5	0	0	6	0	25	0	159
K1	2	9	2,37	4	0	0	5	0	27	0	78
K2	2	8	2,41	1	0	0	4	0	31	0	90
K3	2	8	2,17	6	0	0	1	0	29	0	71
K4	2	10	2,37	0	0	0	7	0	29	0	88
K5	2	9	2,34	4	0	0	3	0	29	0	79
K6	2	11	2,34	5	0	0	4	0	27	0	90
K7	2	10	2,46	3	0	0	4	0	29	0	106
K8	2	8	2,17	7	0	0	4	0	25	0	65
K9	2	8	2,34	6	0	0	4	0	26	0	89
K10	2	10	2,24	5	0	0	4	0	27	0	74

Tabulka 5.36: Parametry obvodu dk15.blif a jeho klonů

Při pohledu na tab. 5.36, kde se nacházejí zbylé parametry originálního obvodu a deseti jeho klonů, je vidět velké průměrné větvení originálů 3.02, kterému se nejvíce přibližuje klon K7 s hodnotou 2.46. Obvody jsou tvořeny převážně z hradel NAND s malým počtem hradel AND a INV. Výsledného poměru bylo dosaženo ve všech obvodech. Poslední sloupec tabulky ukazuje počet základních smyček rekonvergenčí. Originální obvod dosahuje 159 rekonvergenčí, k tomuto číslu se nejvíce přibližuje již zmiňovaný obvod K7, který jich má 106.

Minimalizace takto malých obvodů je jednodušší než u velkých. Největší odolnost vůči minimalizaci ukázal originální obvod, který se zvětšil o 4 hradla, jak je vidět v tabulce 5.37. Největší obvod má klon K5, dosahuje 29 hradel. K největší minimalizaci došlo u K7, kde po minimalizaci zbylo pouze 13 hradel z původních 36. Právě tento obvod dosahoval před minimalizací největšího počtu rekonvergenčí.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	4	40	3	9	2,47	95	111,11
K1	4	21	3	6	1,88	11	58,33
K2	4	26	3	6	2,06	24	72,22
K3	3	19	3	6	1,96	4	52,78
K4	3	27	3	6	2,00	16	75,00
K5	4	29	3	8	2,09	41	80,56
K6	4	22	4	7	2,07	22	61,11
K7	3	13	3	5	1,89	3	36,11
K8	4	20	2	4	1,85	13	55,56
K9	4	24	3	5	2,07	34	66,67
K10	3	23	3	8	1,96	15	63,89

Tabulka 5.37: Parametry po minimalizaci souboru dk15.blif a jeho klonů

5.3.11 Obvod tav.blif

Druhý sekvenční obvod z balíku MCNC se jmenuje tav.blif. Jedná se o obvod s nejmenším počtem hradel, z kterého budou vytvářeny klony. Je tvořen pouze z 19 hradel. Zbylé jeho parametry jsou v tabulce 5.38.

Vstupy	Výstupy	Latch	Úrovně	Hradla
4	4	2	3	19

Tabulka 5.38: Stejně parametry obvodu tav.blif s klony

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	2	4	1,96	5	0	1	1	0	12	0	6
K1	2	4	1,92	2	0	0	0	0	17	0	19
K2	2	4	1,88	4	0	1	2	0	12	0	19
K3	1	4	1,96	0	0	1	1	0	17	0	19
K4	2	4	1,96	2	0	1	0	0	16	0	20
K5	1	4	1,88	2	0	2	0	0	15	0	17
K6	2	4	1,96	1	0	1	1	0	16	0	21
K7	2	4	1,84	3	0	2	0	0	14	0	16
K8	2	4	1,96	0	0	2	1	0	16	0	22
K9	2	4	1,96	0	0	0	2	0	17	0	24
K10	1	4	1,88	3	0	1	1	0	14	0	19

Tabulka 5.39: Parametry obvodu tav.blif a jeho klonů

V tabulce 5.39 jsou vypsané hodnoty proměnných parametrů originálního obvodu a jeho klonů. V tomto případě se jedná o velice přesné klony. Maximální větvení je vždy rovno 4, i průměrné větvení signálu se u klonů rovná nebo velice přibližuje originální hodnotě 1.96. Jediný významný rozdíl se nalézá u počtu rekonvergenzí, kde originální obvod tav.blif má 6 rekonvergenzí a u klonů se tato hodnota pohybuje kolem hodnoty 18.

Minimalizované hodnoty z tabulky 5.40 zobrazují stejné hodnoty parametrů u originálního obvodu jako před minimalizací. Nedošlo tedy k žádnému zmenšení. U klonů bylo největší zmenšení zaznamenáno u K3 a K10 na 13 hradel z původních 19. Klon K6 se zvětšil o jedno hradlo na celkový počet 20. Klony tohoto obvodu dosáhly horší minimalizace než u předchozího sekvenčního obvodu. Důvod lze najít ve větším počtu vstupů, výstupů a také menšímu počtu hradel. Tato situace zamezí větší minimalizaci, protože již původní velikost obvodu se blíží k optimální hodnotě.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	3	19	2	4	1,96	6	100,00
K1	3	17	2	4	1,91	15	89,47
K2	3	13	2	3	1,63	4	68,42
K3	3	17	2	4	1,87	15	89,47
K4	3	18	2	4	1,83	13	94,74
K5	4	17	2	4	1,78	12	89,47
K6	4	20	2	4	1,77	14	105,26
K7	3	16	2	4	1,86	13	84,21
K8	4	18	2	4	1,88	17	94,74
K9	3	18	2	4	1,83	18	94,74
K10	3	13	1	4	1,68	4	68,42

Tabulka 5.40: Parametry po minimalizaci souboru tav.blif a jeho klonů

5.3.12 Obvod train11.blif

Posledním testovaným sekvenčním obvodem je train11.blif. Je zajímavý malým počtem vstupů (2) a výstupů (1). V obvodu se nachází i čtyři klopné obvody. Celkový počet 27 hradel se rozkládá na čtyřech úrovních. Tyto parametry jsou přehledně v tabulce 5.41.

Vstupy	Výstupy	Latch	Úrovně	Hradla
2	1	4	4	27

Tabulka 5.41: Stejně parametry obvodu train11.blif s klony

#	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR	Rek.
Orig.	3	7	2,27	6	0	0	4	0	17	0	63
K1	3	6	2,12	4	0	0	1	0	22	0	49
K2	2	6	1,97	3	0	0	3	0	21	0	41
K3	3	6	1,97	5	0	0	3	0	19	0	36
K4	3	5	2,00	4	0	0	1	0	22	0	42
K5	2	7	2,06	2	0	0	4	0	21	0	41
K6	3	7	2,15	1	0	0	5	0	21	0	43
K7	2	7	2,09	4	0	0	0	0	23	0	40
K8	2	6	1,97	6	0	0	1	0	20	0	54
K9	3	7	2,00	2	0	0	2	0	23	0	34
K10	3	6	2,15	4	0	0	3	0	20	0	53

Tabulka 5.42: Parametry obvodu train11.blif a jeho klonů

Parametry vygenerovaných klonů jsou v tab. 5.42. Opět je zde velice dobré dodržení hodnot originálního obvodu. V počtu maximálního větvení signálu se nejvíce odchýlil klon K4 s hodnotou 5, což je o 2 méně než originál s počtem 7. Průměrné větvení dosahuje maximální hodnoty 2.27 u originálního obvodu. Klony se této hodnotě pouze přibližují a nejbližší je K1 s 2.12. I zde se opakuje vyšší počet smyček rekonvergenzí u originálního obvodu oproti klonům.

Poslední tabulka 5.43 ukazuje hodnoty po minimalizaci syntézním nástrojem. U originálu nedošlo k žádné změně. Klony dosáhly minimálního zmenšení 89 % u K8 a největšího u K7 59 %.

#	Úrovně	Hradla	Návaz.	Max. vět.	Pr. vět.	Rekon.	Velikost [%]
Orig.	4	27	2	5	2,12	63	100,00
K1	4	19	3	6	2,00	28	70,37
K2	4	24	3	6	1,87	23	88,89
K3	4	23	3	5	1,97	30	85,19
K4	4	20	3	6	1,81	22	74,07
K5	3	20	2	6	2,15	33	74,07
K6	4	23	2	6	2,21	36	85,19
K7	3	16	2	5	1,82	11	59,26
K8	4	24	2	5	1,97	36	88,89
K9	4	22	3	5	1,83	24	81,48
K10	4	22	2	6	2,18	37	81,48

Tabulka 5.43: Parametry po minimalizaci souboru train11.blif a jeho klonů

5.3.13 Zhodnocení

Je důležité zdůraznit fakt, že vytvořené klony mají náhodné uspořádání, proto neplní žádnou konkrétní funkci na rozdíl od originálu. Při vytváření klonů nedošlo k žádné selekci, vzniklo pouze deset obvodů, na kterých proběhlo měření. Z výsledků minimalizace vyplývá, že žádný nově vytvořený obvod nebyl zmenšen na absolutní nulu. Stanovená hranice maximální řiditelnosti 90 % se osvědčila.

Tato měření jsou spolu s obvody a klony uložena na příloženém CD ve složce data\klony. Bylo provedeno i měření na souborech c5315.blif, comp.blif a cordic.blif, které se v této práci neobjevilo.

Klony obvodů dosahovaly velice přesně parametrů svých vzorů. Pouze u maximálního větvení signálu se u některých testovacích obvodů objevilo větší číslo než u klonů. Tento jev je způsoben nemožností ovlivnit tento parametr u generovaných obvodů směrem nahoru. Obdobně k tomuto jevu došlo i u průměrného větvení, ne však v takové míře a rozsahu. Poměrné zastoupení typů logických členů bylo také dodrženo.

Z výsledků měření vyplývá, že klony testovacích kombinačních obvodů se hůře minimalizují. Pouze u souboru c1908.blif bylo průměrné procento minimalizace klonů stejné jako u originálu. V tomto případě však tento fakt nehraje významnou roli, protože u originálu došlo k výraznému zvětšení obvodu po minimalizaci, koeficient dosahoval 146 %. Druhý soubor, který se také lišil, je count.blif, který po minimalizaci dosahoval 105 % své původní velikosti. U klonů tohoto testovacího obvodu se hladina pohybovala okolo 94 %.

Klony testovaných sekvenčních obvodů dosáhly větší míry minimalizace než tomu bylo u kombinačních obvodů. Zde hraje roli velikost, která byla velice malá v poměru s kombinačními obvody. Přesto se procentuální zmenšení klonů pohybovalo v průměru na 62 % u obvodu dk15.blif, 88 % u tav.blif a 79 % u train11.blif.

Zajímavým parametrem pro sledování se stal i počet základních smyček rekonvergenčí. Generované obvody dosahovaly téměř vždy většího počtu smyček než vzory. Tento počet se odvíjí od atributů generování obvodu a vzniklé náhodné struktury. Při minimalizaci došlo téměř vždy ke snížení této hodnoty u originálních obvodů, zatímco u klonů došlo ke zvýšení. Jediný obvod, který vybočuje

je c1908.blif s velkým počtem rekonvergenčí, jež daleko přesáhly hodnoty u vygenerovaných obvodů a po minimalizaci vzrostly.

Získáním parametrů testovaných obvodů a následné vytvoření klonů s co nejpodobnějšími parametry vede k obdobné, či vyšší odolnosti vůči minimalizačním technikám.

5.4 Shrnutí dodržování požadovaných parametrů obvodu

Na úplný závěr přichází shrnutí dodržení parametrů při generování obvodu. Parametry lze rozdělit z tohoto hlediska do tří kategorií.

Do první skupiny patří ty, které jsou dodrženy vždy. Jejich plnění je stanoveno principy generování a tudíž, pokud je obvod vytvořen, je dosaženo přesného splnění zadání. V této skupině jsou základní parametry obvodu.

- Počet vstupních signálů
- Počet výstupních signálů
- Úroveň vzniku jednotlivých výstupních signálů
- Počet klopných obvodů
- Počet úrovní obvodu mezi vstupem a výstupem jednotlivých klopných obvodů
- Celkový počet úrovní obvodu
- Počet logických prvků

Druhou kategorií jsou parametry, které omezují vlastnosti obvodu stanovením horní hranice. U maximální návaznosti signálů a maximálního počtu vstupů hradla závisí na náhodném generátoru čísel, zda jednotlivý parametr dosáhne oné horní hranice. Nastavení četnosti typů hradel je dodržováno také při jednotlivém vytváření hradel, tudíž výsledné počty odpovídají definované četnosti.

- Maximální délka signálu neboli návaznost
- Maximální počet vstupů hradla
- Četnost jednotlivých typů hradel

Poslední třetí skupinou jsou parametry, které se upravují až po vytvoření obvodu. Algoritmus úpravy počítá se zmenšováním hodnot a nemusí být vždy úspěšný. Průměrné větvení signálu závisí na samotné struktuře obvodu definované předešlými parametry, jeho jeho snížení lze docílit pouze snižováním počtu vstupů logických hradel, pokud existují hradla s třemi a více vstupy.

- Maximální větvení signálu
- Průměrné větvení signálů

Počet rekonverzí v obvodu je věc, kterou nelze nijak ovlivnit při náhodném generování obvodu. Je dána celkovou strukturou generovaného obvodu podle všech výše zmíněných parametrů.

V následující tabulce 5.44 jsou zobrazeny průměrné hodnoty procentuálních odchylek parametrů u všech testovaných standardních obvodů. Jsou zde pouze parametry, u kterých není 100 % jistota, že budou splněny. Pro každý klon obvodu byla vypočítána procentuální odchylka parametrů vůči původní hodnotě. Pro soubor Apex6.blif byla původní hodnota návaznosti signálů 11, klon K1 dosáhl hodnoty 10, což dává procentuální odchylku $100 - \frac{10}{0.11} = 9.1\%$. V tabulce 5.44 jsou průměrné hodnoty ze všech deseti vytvořených klonů jednotlivých obvodů.

V tabulce 5.45 jsou naopak vypsány průměrné absolutní hodnoty odchylky od potřebné hodnoty.

Tyto dvě tabulky se navzájem doplňují, protože ve většině případů se může zdát procentuální odchylka velká, ale absolutní odchylka se ukáže jako malá. Rozdíl je způsoben malou původní hodnotou, se kterou se tyto hodnoty porovnávají. Tato situace nastává při největší procentuální odchylce, kterou je možno nalézt v tabulce. Jedná se o soubor Apex7.blif, který obsahuje pouze jeden NXOR. Klony byly vytvořeny s rozsahem výskytu tohoto hradla od 0 do 4. Průměrné procentuální odchýlení je zde 120 %, ale v tabulce s průměrnými absolutními hodnotami je 1.2 .

Procentuální průměrná odchylka [%]										
S.	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR
Apex6	9,1	62,4	9,36	7,5	85,0	95,0	26,0	51,7	8,0	64,4
Apex7	0,8	68,8	8,82	10,6	120,0	0,0	24,0	46,0	8,8	57,3
b9	0,0	52,7	3,41	22,4	0,0	0,0	25,0	17,5	18,4	40,0
C880	0,0	20,0	0,02	16,7	62,2	0,0	24,7	41,0	3,2	41,6
C1908	2,0	11,8	7,43	12,4	47,5	59,5	38,0	56,7	26,8	30,0
C5315	1,4	58,5	0,00	14,2	65,1	59,5	28,5	54,0	5,1	55,8
comp	7,3	0,0	0,34	17,9	0,0	71,5	26,7	0,0	16,2	45,0
cordic	14,3	10,0	0,50	4,7	92,5	80,0	46,7	30,0	8,4	44,0
count	1,4	74,4	0,46	15,8	0,0	0,0	32,0	24,5	15,4	52,0
term1	7,8	5,5	0,09	30,0	0,0	0,0	28,1	34,4	6,7	62,0
too-large	8,7	56,3	0,09	24,3	0,0	0,0	20,0	50,0	10,5	68,1
dk15	0,0	17,3	23,23	34,0	0,0	0,0	36,7	0,0	11,6	0,0
tav	15,0	0,0	2,04	66,0	0,0	50,0	60,0	0,0	28,3	0,0
train11	13,3	10,0	9,87	41,7	0,0	0,0	47,5	0,0	24,7	0,0
Průměr	5,8	32,0	4,69	22,7	33,7	29,7	33,1	29,0	13,7	40,0

Tabulka 5.44: Průměrná procentuální odchylka parametrů

Při pohledu na maximální návaznost signálů (Náv.), jsou zřejmé dvě větší průměrné procentuální odchylky. Obvody cordic.blif a tav.blif mají sice velkou procentuální odchylku 14.3 % a 15 %, ale malou absolutní 1 a 0.3. Tento parametr se daří velice dobře dodržet.

Maximální větvení signálu (M. v.) je hodně ovlivněno obvody s velkou hodnotou tohoto parametru. Například obvod count.blif dosahuje hodnoty 32, což znamená, že existuje signál, který vede do 32 hradel, což je při velikosti obvodu (116) téměř třetina. Náhodně generované obvody této hodnoty nedosáhly.

U průměrného větvení signálu (Pr. v.) bylo dosaženo velice přesného splnění požadavků, které

Absolutní průměrná odchylka [-]										
S.	Náv.	M. v.	Pr. v.	INV	XOR	NXOR	AND	OR	NAND	NOR
Apex6	1,0	15,6	0,17	3,8	1,7	3,8	7,8	3,1	36,6	29,0
Apex7	0,1	17,9	0,16	3,4	1,2	0,0	1,2	2,3	13,3	12,6
b9	0,0	5,8	0,05	4,7	0,0	0,0	1,0	0,7	9,0	4,4
C880	0,0	1,8	0,00	3,5	11,2	0,0	4,7	4,1	4,8	7,9
C1908	0,3	1,3	0,15	3,6	24,7	13,1	13,3	1,7	21,4	4,8
C5315	0,3	11,7	0,00	19,3	72,3	11,9	16,8	5,4	35,0	46,9
comp	0,8	0,0	0,01	3,4	0,0	9,3	3,2	0,0	6,8	0,9
cordic	1,0	0,3	0,01	0,8	3,7	1,6	1,4	1,2	2,7	2,2
count	0,1	23,8	0,01	3,0	0,0	0,0	1,6	5,4	10,0	2,6
term1	0,7	0,6	0,00	9,3	0,0	0,0	5,9	3,1	8,0	9,3
too-large	2,0	13,5	0,00	11,2	0,0	0,0	4,8	2,0	34,3	39,5
dk15	0,0	1,9	0,70	1,7	0,0	0,0	2,2	0,0	2,9	0,0
tav	0,3	0,0	0,04	3,3	0,0	0,5	0,6	0,0	3,4	0,0
train11	0,4	0,7	0,22	2,5	0,0	0,0	1,9	0,0	4,2	0,0
Průměr	0,5	6,8	0,11	5,3	8,2	2,9	4,7	2,1	13,7	11,4

Tabulka 5.45: Průměrná absolutní odchylka parametrů

je patrné z pohledu na jednotlivé průměrné absolutní odchylky. V devíti případech ze čtrnácti byla absolutní průměrná odchylka pod hranicí 0.05 .

Následují odchylky v četnosti vytváření přesného počtu hradel daných typů. Výsledné počty reflektují princip volby typu hradla při generování. Jedná se o náhodný výběr, kde je zohledněna potřebná četnost typů. Proto dochází k mírnému odchýlení, avšak potřebné poměry jsou dodrženy.

Kapitola 6

Závěr

Cílem diplomové práce bylo vytvořit parametrizovaný generátor logických obvodů, který bude sloužit pro testování syntézních nástrojů. Vytvořený program generuje obvody v závislosti na počtu vstupních a výstupních signálů, počtu hradel a úrovní obvodu, maximální délky signálů a maximálního počtu vstupů hradla. Tyto parametry jsou dodrženy vždy spolu s počtem úrovní mezi vstupy a jednotlivým výstupem a také četností výskytu jednotlivých typů hradel.

Maximální a průměrné větvení signálu je následně upraveno na hodnoty co nejvíce splňující další vstupní parametry. Na závěr generování se vypočte množství základních smyček rekonvergenčí.

Jako další možnost při generování je tvorba sekvenčního obvodu, definovaná pomocí počtu klopných obvodů typu D a množství úrovní mezi jeho vstupem a výstupem.

Vzniklé obvody byly testovány pomocí syntézního nástroje ABC na minimalizovatelnost. Výsledky měření prokázaly velice špatnou schopnost obvodů být minimalizován. Při generování vzniká také velké množství smyček rekonvergenčí, což je dáno náhodným generováním a výslednou strukturou obvodu závislou na parametrech.

Dále byly vytvořeny klony standardních testovacích obvodů za účelem porovnání vlastností. Klony, co nejpřesněji splňující parametry originálu, dokázaly minimálně vyrovnat poměr zmenšení originálu. Ve většině případů však došlo ke zvýšení odolnosti vůči minimalizaci.

Zadání jsem bez výhrad splnil a vytvořil jsem program pro náhodné generování víceúrovňových obvodů.

Literatura

- [1] BRGLEZ, F. On testability of combinational networks. *Circuits and Systems*. 1984, s. 221–225.
- [2] FIŠER, P. BOOM-II the Boolean Minimizer. Dostupné z: <<http://service.felk.cvut.cz/vlsi/prj/BOOM/>>.
- [3] MĚCHURA, T. Generátor náhodných logických obvodů. 2008. Dostupné z: <<http://service.felk.cvut.cz/vlsi/dip/Mechura08/texty/dipl.pdf>>. Diplomová práce.
- [4] web:berkeley.edu. ABC: A System for Sequential Synthesis and Verification, . Dostupné z: <<http://www.eecs.berkeley.edu/~alanmi/abc/>>.
- [5] web:berkeley.edu. The Donald O. Pederson Center for Electronic Systems Design, . Dostupné z: <http://embedded.eecs.berkeley.edu/Alumni/teh/research/papers/blif_spec_92.pdf>.
- [6] web:cplusplus. cplusplus.com - The C++ Resources Network. <http://cplusplus.com/>, stav z 15. 3. 2012.
- [7] YANG, S. Logic Synthesis and Optimization Benchmarks User Guide Version 3.0, 1991.

Příloha A

Naměřená data

Šířka mřížky	Počet úrovní obvodu						
	10	20	30	40	50	60	70
10	55,3778	141,422	209,089	317,689	369,333	456,4	534,6
15	68,7048	188,829	300,886	396,219	552,333	662,514	805,124
20	85,6053	256,495	396,105	556,532	721,126	834,316	972,953
30	118,163	359,345	579,832	803,292	1047,94	1272,57	1508,87
40	143,764	434,197	738,1	993,764	1370,16	1687,17	1993,19
50	155,371	533,38	935,66	1306,01	1688,77	2078,6	2469,53
60	182,915	588,252	1073,93	1550,49	2007	2431,38	2929,01
70	200,707	711,019	1243,78	1743,8	2321,38	2852,06	3363,19
80	219,14	782,546	1402,61	2016,55	2663,31	3233,06	3812,21
90	235,764	883,094	1559,48	2292,65	2912,95	3577,3	4303,24
100	234,521	985,832	1731,95	2514,9	3236,66	3993,57	4706,9
120	263,412	1152,7	2048,97	2959,07	3845,09	4759,98	5700,48
140	288,472	1303,46	2357,82	3390,9	4454,51	5503,69	6530,54
150	290,108	1386,03	2542,74	3670,82	4807,9	5883,85	7069,88

Tabulka A.1: Závislost průměrného počtu hradel na šířce mřížky graf 3.4

Počet vstupů	4
Návaznost signálů	1
Maximální počet vstupů hradla	2

Tabulka A.2: Parametry k tab A.1

Šířka mřížky	Počet vstupů			
	5 vstupů	10 vstupů	15 vstupů	20 vstupů
20	384,521	412,521	389,863	399,716
30	571,639	579,143	609,917	590,126
40	746,513	780,913	729,456	754,309
50	905,528	877,82	934,407	902,479
60	1112,59	1081,38	1082,07	1080,96
70	1231,49	1224,26	1243,64	1241,08
80	1374,88	1410,97	1435,42	1405,84
100	1720,96	1708	1784,73	1672,17
150	2479,69	2495,38	2533,57	2506,8

Tabulka A.3: Data, závislost na počtu vstupních signálů graf 3.5

Počet úrovní obvodu	30
Návaznost signálů	1
Maximální počet vstupů hradla	2

Tabulka A.4: Parametry k tab A.3

Šířka mřížky	Maximální návaznost signálů					
	N=1	N=2	N=5	N=10	N=15	N=20
30	359,35	327,46	260,23	187,13	152,42	148,42
40	434,20	406,93	316,95	199,61	185,83	168,96
50	533,38	498,39	364,34	261,01	227,69	198,85
60	588,25	562,73	402,69	267,65	242,55	218,63
70	711,02	651,19	480,94	327,26	260,44	243,30
80	782,55	706,52	514,94	340,05	268,30	261,05
90	883,09	780,82	553,30	370,93	277,31	264,95
100	985,83	845,30	587,68	386,03	295,27	283,62
120	1152,70	1007,35	686,76	425,64	329,72	317,01
140	1303,46	1136,90	751,93	455,84	371,28	338,82
150	1386,03	1209,89	782,94	482,54	376,14	357,99

Tabulka A.5: Data, závislost na návaznosti vstupních signálů graf 3.6

Počet vstupů	5
Počet úrovní obvodu	20
Maximální počet vstupů hradla	2

Tabulka A.6: Parametry k tab A.5

Šířka mřížky	Max počet vstupu hradla				
	2	3	4	5	6
30	359,35	375,39	386,36	389,26	401,00
40	434,20	492,49	497,37	508,93	533,53
50	533,38	592,78	623,57	642,14	630,23
60	588,25	703,78	737,49	742,93	741,78
70	711,02	800,73	857,97	856,21	871,70
80	782,55	893,91	958,44	968,94	973,28
90	883,09	991,56	1077,78	1087,33	1068,86
100	985,83	1130,58	1177,84	1198,58	1188,71

Tabulka A.7: Data, závislost na maximálním počtu vstupů hradla graf 3.7

Počet vstupů	5
Počet úrovní obvodu	20
Návaznost vstupních signálů	1

Tabulka A.8: Parametry k tab A.7

Šířka mřížky	20 úrovní		30 úrovní		50 úrovní	
	N1	N5	N1	N5	N1	N5
3	50,67	46,20	77,00	71,67	139,33	117,40
5	76,10	61,50	123,80	100,28	204,80	179,42
7	99,71	79,86	158,76	136,91	290,19	243,17
10	126,47	103,10	206,69	181,53	372,20	331,19

Tabulka A.9: Data, počáteční bod charakteristiky graf 3.8

Počet vstupů	5
Maximální počet vstupů hradla	2

Tabulka A.10: Parametry k tab A.9

	Vstupy	Úrovně	Max návaz.	Vst. hradla	Šířka mřížky	Min	Průměr	Max
M1	5	20	2	2	80	564	706,52	802
M2	5	25	15	2	80	262	458,434	717
M3	5	30	15	2	80	426	637,71	957
M4	5	45	10	2	80	1366	1691,03	1971
M5	5	60	40	2	80	1139	1518,96	2039

Tabulka A.11: Rozptyl kolem průměru hodnot

Levels	Počet hradel po minimalizaci									
	20			30			40			
	PI	5	10	15	5	10	15	5	10	15
1	234	262	288	650	787	790	845	1018	904	
2	239	284	289	668	772	777	788	977	970	
3	251	280	290	697	760	789	876	977	973	
4	261	293	283	667	744	764	873	966	995	
5	242	281	291	667	788	747	849	962	970	
6	262	266	286	663	766	804	912	982	975	
7	237	274	286	619	769	762	917	995	998	
8	210	257	262	660	763	770	752	963	967	
9	246	292	293	638	783	758	936	983	975	
10	186	263	276	680	760	771	960	969	973	

Tabulka A.12: Data testování závislosti počtu vstupů při PO=5, N=10, vst. hradla=2

Levels	Počet hradel po minimalizaci									
	PO	20			30			40		
		1	5	10	1	5	10	1	5	10
1	274	262	288	728	787	759	974	1018	973	
2	279	284	273	761	772	783	977	977	957	
3	279	280	266	752	760	774	973	977	981	
4	289	293	285	770	744	766	888	966	997	
5	284	281	270	778	788	788	971	962	988	
6	288	266	273	749	766	756	779	982	965	
7	280	274	284	742	769	774	934	995	990	
8	272	257	280	769	763	761	963	963	970	
9	286	292	289	718	783	771	984	983	987	
10	279	263	273	768	760	773	984	969	975	

Tabulka A.13: Data testování závislosti počtu výstupů při PI=10, N=10, vst. hradla=2

Levels	Počet hradel po minimalizaci									
	20			30			40			
	1	10	20	1	10	20	1	10	20	40
1	283	262	261	727	787	740	937	1018	970	947
2	297	284	276	728	772	758	925	977	945	938
3	278	280	278	762	760	764	963	977	912	931
4	260	293	269	741	744	776	936	966	925	953
5	263	281	288	740	788	723	960	962	940	926
6	243	266	283	754	766	738	913	982	980	966
7	284	274	265	739	769	749	917	995	930	953
8	255	257	273	727	763	780	948	963	964	958
9	282	292	280	677	783	690	968	983	1012	947
10	278	263	278	731	760	733	938	969	965	948

Tabulka A.14: Data testování závislosti na návaznosti signálů při PI=10, PO=5, vst. hradla=2

Levels	Počet hradel po minimalizaci								
	20			30			40		
	2	3	4	2	3	4	2	3	4
1	262	395	484	787	1184	1224	1018	1534	1582
2	284	432	419	772	1158	1229	977	1591	1449
3	280	368	405	760	1128	1282	977	1533	1617
4	293	400	419	744	1112	1205	966	1498	1664
5	281	399	442	788	1191	1290	962	1475	1614
6	266	413	425	766	1125	1287	982	1601	1555
7	274	408	398	769	1051	1304	995	1523	1590
8	257	396	451	763	1101	1307	963	1663	1661
9	292	395	411	783	1227	1134	983	1435	1722
10	263	385	460	760	1126	1274	969	1498	1558

Tabulka A.15: Data testování závislosti max. počtu výstupů hradla při PI=10, PO=5, N=10

Levels	Počet hradel po minimalizaci					
	20		30		40	
	0	1	0	1	0	1
XOR						
1	219	262	637	787	786	1018
2	251	284	634	772	779	977
3	235	280	613	760	829	977
4	206	293	667	744	814	966
5	234	281	585	788	798	962
6	236	266	608	766	824	982
7	170	274	609	769	786	995
8	211	257	608	763	804	963
9	206	292	569	783	852	983
10	232	263	642	760	867	969

Tabulka A.16: Data testování výskytu hradel XOR při PI=10, PO=5, N=10, vst. hradla=2, vst. hradla=2

Levels	Počet hradel po minimalizaci								
	20			30			40		
	PI	5	10	15	5	10	15	5	10
1	309	403	462	998	1130	1310	1072	1544	1585
2	340	496	452	783	1303	1303	1158	1478	1554
3	317	320	492	975	1247	1219	1080	1538	1584
4	308	464	423	984	1276	1197	1152	1610	1704
5	338	359	475	800	1354	1243	1248	1522	1757
6	264	441	461	1006	1161	1284	1285	1613	1649
7	345	431	357	871	1238	1273	1229	1589	1723
8	340	463	460	864	1096	1222	1408	1686	1513
9	283	503	452	926	1254	1145	1261	1528	1592
10	317	457	423	957	1287	1313	1085	1715	1682

Tabulka A.17: Data testování nejhorší možnosti nastavení při PO=10, N=10, vst. hradla=4, s XOR

Příloha B

Formát BLIF

Formát BLIF (Berkeley Logic Interchange Format) je již standardní textový formát pro zápis víceúrovňových logických obvodů. V této příloze je jeho základní popis, podrobnější informace najdete zde [5].

Na začátku každého obvodu se definuje jméno modelu.

```
.model [název]
```

Poté následují jména vstupních signálů, která jsou oddělena mezerou.

```
.inputs pi1 pi2 pi3 ... pin
```

Obdobně je tomu u výstupních signálů.

```
.outputs po1 po2 po3 ... pon
```

Následuje volitelná část definující latch. První je jméno vstupního signálu, následně výstupního a na konec počáteční hodnota.

```
.latch vstup vystup 0
```

Definice logických prvků sítě začínají hlavičkou, ve které se určí množina vstupních signálů a jeden výstupní, který je odlišen pouze svoji pozicí, je na konci. Tento formát nepodporuje vícevýstupová hradla. Následuje specifikace funkce logického prvku. Jedná se o množinu termů, které společně specifikují nulu nebo jedničku na výstupu. Termy jsou logické výrazy zadané disjunktivní formou. Výraz $x_1 \cdot \bar{x}_2 = 1$ se zapíše jako `10 1`. Vždy se musí uvést hodnoty všech vstupních signálů. Některý vstupní signál nemusí být v určitém termu specifikován, zde se použije na jeho místo znak `-`.

```
.names a b c x
110 0
1-1 0
```

Zde je vidět definice třívstupového prvku (a, b, c) s výstupem x . Skládá se ze dvou termů definujících na výstupu nulu. Zápis odpovídá výrazu $a \cdot b \cdot \bar{c} + a \cdot c = 0$.

Je také možné definovat konstanty.

```
.names k
1
```

Logické členy se zapisují pod sebe a celý soubor je zakončen `.end`.

```
.end
```

Příloha C

Instalační a uživatelská příručka

C.1 Instalace

Pro operační systém Windows je nutné pouze zkopírovat složku s již zkompilevaným programem na libovolné místo na disku a možné začít program používat. Je důležité zkontrolovat, zda adresář s programem obsahuje soubor s daty pro heuristiku `heur.dat`.

Pokud se jedná u unixové systému je nutné program zkompilevat. To se provede zkopírováním složky `source` na disk a spuštění `makefile` ve složce `source\BOOM\Releases\Linux\` příkazem `make`. Ve složce `bin` se vytvoří soubor `Gener`, který dle svých potřeb spolu se souborem `heur.dat` přesunete na vámi požadované místo.

C.2 Uživatelská příručka

První funkcionalita programu je zjištění parametrů již existujícího obvodu Blif [5]. Tento soubor musí být přemapovaný pomocí programu ABC [4]. Podrobně je to popsáno v kapitole 4.12.

První parametr je soubor Blif a druhý nepovinný je název textového souboru pro uložení parametrů do formátu `pro vstup` do tohoto programu. Když tento parametr nebude obsažen, soubor se uloží pod názvem `param.txt`. Dále se také vytvoří soubor s informacemi s příponou `txt` a názvem schodným s modelem vstupního obvodu.

```
Gener[.exe] -info obvod.blif [paramObv.txt]
```

Hlavní funkcí programu je generování obvodů, to se spouští následovně.

```
Gener[.exe] param.txt
```

Jediný parametr je textový soubor následujícího formátu.

C.2.1 Formát vstupního souboru

Textový formát se skládá ze tří částí podle důležitosti parametrů.

První část má specifikovat základní parametry obvodu a jsou povinné. Tím jsou počet název obvodu, počet vstupních signálů, výstupních a počet úrovní obvodu. Pouze tyto parametry stačí k vytvoření obvodu.

```
.model
obvod
.inputs
5
.outputs
3
.levels
10
```

Další možností je specifikování počtu klopných obvodů, které jsou defaultně nastaveny na nula.

```
.latch
2
```

Určení počtu logických hradel v obvodu.

```
.nodes
60
```

Při definování počtu hradel je možné nastavit i počet úrovní mezi vstupy a jednotlivými výstupy, jako počet úrovní hradel mezi vstupem a výstupem klopného obvodu. Při použití těchto parametrů je nutné definovat všechny výstupy respektive úrovně u latch a také aby minimálně jedna hodnota u těchto parametrů byla rovna celkovému počtu úrovní `.levels`. Při nepoužití těchto parametrů jsou hodnoty nastaveny na `.levels`.

```
.leveloutput
10 9 8 7 2
.levellatch
2 2
```

Následuje nastavení maximálního počtu vstupů hradla, defaultně nastaveno na dva, což je minimální hodnota.

```
.maxinputsnode
3
```

Maximální délka vstupních signálů neboli návaznost, v základu nastavená na jedna.

```
.maxlengthsignals
6
```

V druhé části je stanovení omezení parametrů větvení. Průměrné větvení signálu:

```
.averagesignalfanout  
1.8
```

Maximální větvení signálu:

```
.maxsignalfanout  
6
```

Pokud nejsou hodnoty větvení uvedeny, neprovede se žádná změna větvení.

Poslední informativní část slouží k zakázání výpočtu rekonvergenčí, protože tato část algoritmu je časově nejnáročnější a u větších souborů může trvat delší dobu, přitom ani informace o rekonvergencích není potřeba znát.

```
.reconverg  
0
```

Jako další je možné zakázat vytvoření souboru s informacemi o vygenerovaném obvodu, který má stejný název jako obvod pouze s příponou `log`.

```
.log  
0
```


Příloha D

Obsah příloženého CD

index.html - Jednoduchý průvodce CD.

adresář **data**

adresář **klony** - Obsahuje výsledky měření, původní obvody a klony z kapitoly 5.3.

adresář **minimalizace** - Obsahuje výsledky měření z kapitoly 5.2.

adresář **rekonvergence** - Obsahuje výsledky měření z kapitoly 5.1.

adresář **execute** - Zde je zkompileovaný exe soubor generátoru se souborem pro heuristiku. Nachází se zde také ukázka vstupního souboru s parametry.

adresář **genlib** - Adresář obsahuje knihovny hradel pro přemapování nástrojem ABC.

adresář **source** - Adresář zdrojových kódů obsahuje také jádro BOOM, které je nutné pro kompilaci. Nachází se zde projekt Microsoft Visual Studio 2010 pro kompilaci pod Windows a také makefile pro unixové systémy.

adresář **text** - Text diplomové práce v pdf.